

C#
AND ALGORITHMIC THINKING
FOR THE COMPLETE BEGINNER
Second Edition

The Answers

Aristides S. Bouras

Copyright ©

Copyright © by Aristides S. Bouras
<https://www.bouraspage.com>

RCode: 221008

All rights reserved. No part of this book may be reproduced or transmitted in any form or by any means, mechanical or electronic, including photocopying, recording, or by any information storage and retrieval system, without written permission from the authors.

Warning and Disclaimer

This book is designed to provide the answers to all of the review questions, as well as the solutions to all review exercises of the book “C# AND ALGORITHMIC THINKING FOR THE COMPLETE BEGINNER – Second Edition”. Every effort has been taken to make this book compatible with all releases of C#, and it is almost certain to be compatible with any future releases of it.

The information is provided on an “as is” basis. The authors shall have neither liability nor responsibility to any person or entity with respect to any loss or damages arising from the information contained in this book or from the use of the files that may accompany it.

Table of Contents

How to Report Errata.....	7
Chapter 1.....	8
1.7 Review Questions: True/False.....	8
1.8 Review Questions: Multiple Choice.....	8
Review in “Introductory Knowledge”	9
Review Crossword Puzzles	9
Chapter 4.....	11
4.16 Review Questions: True/False	11
4.17 Review Questions: Multiple Choice.....	11
Chapter 5.....	12
5.8 Review Questions: True/False.....	12
5.9 Review Questions: Multiple Choice.....	12
5.10 Review Exercises	12
Chapter 6.....	13
6.4 Review Questions: True/False.....	13
6.5 Review Questions: Multiple Choice.....	13
Chapter 7.....	14
7.7 Review Questions: True/False.....	14
7.8 Review Questions: Multiple Choice.....	14
7.9 Review Exercises	14
Chapter 8.....	15
8.2 Review Questions: True/False.....	15
8.3 Review Exercises	15
Chapter 9.....	17
9.6 Review Exercises	17
Review in “Getting Started with C#”	19
Review Crossword Puzzles	19
Chapter 10	20
10.2 Review Exercises	20
Chapter 11	26
11.3 Review Questions: True/False	26
11.4 Review Questions: Multiple Choice.....	26
11.5 Review Exercises	26

Chapter 12	28
12.2 Review Exercises	28
Chapter 13	31
13.2 Review Exercises	31
Chapter 14	34
14.4 Review Questions: True/False	34
14.5 Review Questions: Multiple Choice	34
14.6 Review Exercises	35
Review in “Sequence Control Structures”	37
Review Crossword Puzzle	37
Chapter 15	38
15.9 Review Questions: True/False	38
15.10 Review Questions: Multiple Choice	38
15.11 Review Exercises	38
Chapter 16	40
16.2 Review Questions: True/False	40
16.3 Review Questions: Multiple Choice	40
16.4 Review Exercises	40
Chapter 17	47
17.2 Review Questions: True/False	47
17.3 Review Questions: Multiple Choice	47
17.4 Review Exercises	47
Chapter 18	55
18.2 Review Questions: True/False	55
18.3 Review Exercises	55
Chapter 19	64
19.2 Review Questions: True/False	64
19.3 Review Exercises	64
Chapter 20	72
20.3 Review Questions: True/False	72
20.4 Review Exercises	72
Chapter 21	78
21.4 Review Exercises	78
Chapter 22	84
22.9 Review Questions: True/False	84

22.10 Review Questions: Multiple Choice	84
22.11 Review Exercises	85
Chapter 23	89
23.6 Review Exercises	89
Review in “Decision Control Structures”	109
Review Crossword Puzzle	109
Chapter 24	110
24.3 Review Questions: True/False	110
Chapter 25	111
25.4 Review Questions: True/False	111
25.5 Review Questions: Multiple Choice	111
25.6 Review Exercises	111
Chapter 26	123
26.3 Review Questions: True/False	123
26.4 Review Questions: Multiple Choice	123
26.5 Review Exercises	123
Chapter 27	137
27.3 Review Questions: True/False	137
27.4 Review Questions: Multiple Choice	137
27.5 Review Exercises	137
Chapter 28	146
28.8 Review Questions: True/False	146
28.9 Review Questions: Multiple Choice	146
28.10 Review Exercises	146
Chapter 29	151
29.4 Review Exercises	151
Chapter 30	160
30.7 Review Questions: True/False	160
30.8 Review Exercises	160
Review in “Loop Control Structures”	183
Review Crossword Puzzle	183
Chapter 31	184
31.13 Review Questions: True/False	184
31.14 Review Questions: Multiple Choice	184
31.15 Review Exercises	184

Chapter 32	198
32.7 Review Questions: True/False	198
32.8 Review Questions: Multiple Choice	198
32.9 Review Exercises	199
Chapter 33	211
33.7 Review Questions: True/False	211
33.8 Review Questions: Multiple Choice	211
33.9 Review Exercises	211
Chapter 34	221
34.7 Review Questions: True/False	221
34.8 Review Exercises	222
Review in “Data Structures in C#”	259
Review Crossword Puzzle	259
Chapter 35	260
35.4 Review Questions: True/False	260
Chapter 36	261
36.8 Review Questions: True/False	261
36.9 Review Exercises	261
Chapter 37	274
37.10 Review Questions: True/False	274
37.11 Review Exercises	274
Chapter 38	284
38.3 Review Exercises	284
Review in “Subprograms”	300
Review Crossword Puzzle	300
Chapter 39	301
39.9 Review Questions: True/False	301
39.10 Review Exercises	301
Review in “Object Oriented Programming”	314
Review Crossword Puzzle	314
Some Final Words from the Author	315

How to Report Errata

Although I have taken great care to ensure the accuracy of the content of this book, mistakes do occur. If you find a mistake in this book, either in the text or the code, I encourage you to report it to me. By doing so, you can save other readers from frustration and, of course, help me to improve the next release of this book. If you find any errata, please feel free to report them by visiting the following address:

<https://www.bouraspage.com/report-errata>

Once your errata are verified, your submission will be accepted and the errata will be uploaded to my website, and added to any existing list of errata.

Chapter 1

1.7 Review Questions: True/False

- | | |
|----------|-----------|
| 1. false | 12. false |
| 2. false | 13. false |
| 3. true | 14. false |
| 4. false | 15. true |
| 5. false | 16. true |
| 6. true | 17. false |
| 7. true | 18. false |
| 8. false | 19. true |
| 9. false | 20. false |
| 10. true | 21. false |
| 11. true | 22. true |

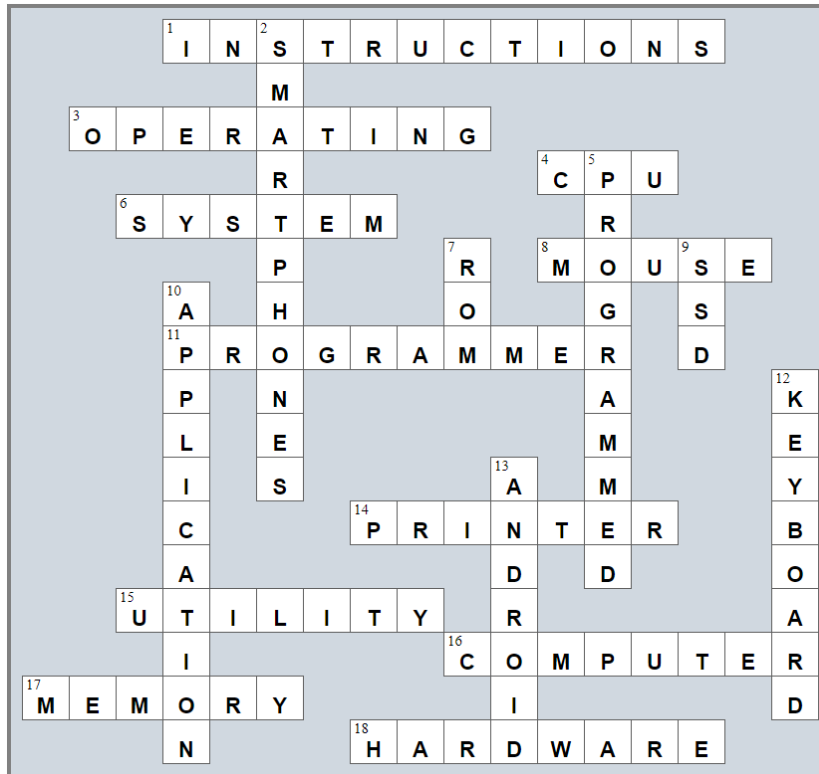
1.8 Review Questions: Multiple Choice

- | | |
|------|-------|
| 1. b | 7. c |
| 2. d | 8. b |
| 3. b | 9. c |
| 4. c | 10. b |
| 5. f | 11. a |
| 6. d | |

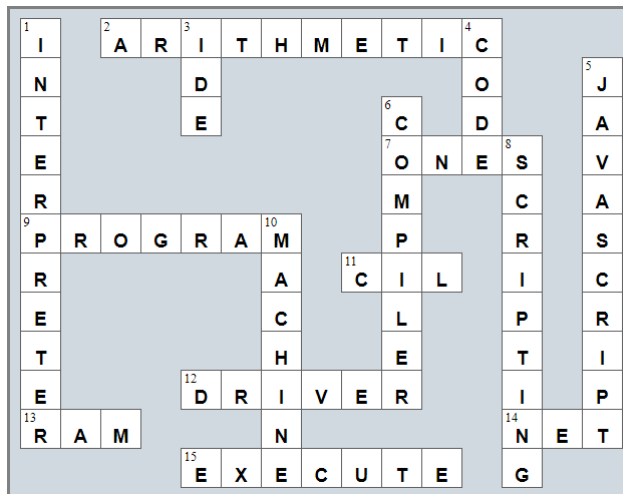
Review in “Introductory Knowledge”

Review Crossword Puzzles

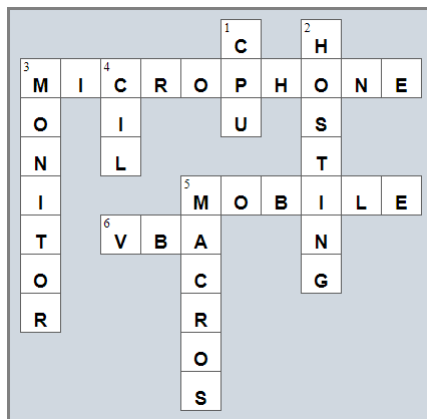
1.



2.



3.



Chapter 4

4.16 Review Questions: True/False

- | | |
|-----------|-----------|
| 1. true | 22. true |
| 2. false | 23. false |
| 3. false | 24. false |
| 4. false | 25. false |
| 5. false | 26. false |
| 6. true | 27. true |
| 7. false | 28. true |
| 8. true | 29. false |
| 9. true | 30. false |
| 10. true | 31. false |
| 11. false | 32. false |
| 12. false | 33. false |
| 13. true | 34. true |
| 14. true | 35. false |
| 15. false | 36. false |
| 16. true | 37. false |
| 17. false | 38. false |
| 18. false | 39. true |
| 19. false | 40. true |
| 20. true | 41. false |
| 21. true | |

4.17 Review Questions: Multiple Choice

- | | |
|------|-------|
| 1. c | 6. a |
| 2. b | 7. b |
| 3. c | 8. d |
| 4. a | 9. a |
| 5. a | 10. d |

Chapter 5

5.8 Review Questions: True/False

- | | |
|----------|-----------|
| 1. false | 10. false |
| 2. false | 11. true |
| 3. true | 12. false |
| 4. false | 13. true |
| 5. false | 14. true |
| 6. true | 15. true |
| 7. false | 16. true |
| 8. false | 17. false |
| 9. true | |

5.9 Review Questions: Multiple Choice

- | | |
|------|------|
| 1. e | 5. c |
| 2. a | 6. c |
| 3. d | 7. d |
| 4. b | |

5.10 Review Exercises

- 1 - c, 2 - d, 3 - a, 4 - b
- 1 - d, 2 - c, 3 - b, 4 - a
-

Value	Data Type	Declaration and Initialization
The name of my friend	String	<code>string name = "Mark";</code>
My address	String	<code>string address = "254 Lookout Rd. Wilson, NY 27893";</code>
The average daily temperature	Float	<code>double average = 70.3;</code>
A telephone number	String	<code>string phone_number = "1-891-764-2410";</code>
My Social Security Number (SSN)	String	<code>string ssn = "123-45-6789";</code>
The speed of a car	Float	<code>double speed = 90.5;</code>
The number of children in a family	Integer	<code>int children = 3;</code>

Chapter 6

6.4 Review Questions: True/False

1. true
2. true
3. true
4. false
5. false

6.5 Review Questions: Multiple Choice

1. a
2. c
3. b
4. b

Chapter 7

7.7 Review Questions: True/False

- | | |
|-----------|-----------|
| 1. false | 14. false |
| 2. true | 15. false |
| 3. false | 16. true |
| 4. false | 17. false |
| 5. false | 18. true |
| 6. false | 19. false |
| 7. false | 20. false |
| 8. false | 21. false |
| 9. true | 22. true |
| 10. false | 23. false |
| 11. false | 24. false |
| 12. true | |
| 13. false | |

7.8 Review Questions: Multiple Choice

- | | | |
|------|------|------|
| 1. c | 4. d | 7. d |
| 2. c | 5. b | 8. c |
| 3. b | 6. d | |

7.9 Review Exercises

- ii, iv, v, ix, x
- i. String, ii. Boolean, iii. String, iv. String, v. Float, vi. Integer
- i. d, ii. f, iii. c, iv. e
- i. 26, ii. 28
- i. 5, ii. 6
- i. 1, ii. 0, iii. 1, iv. 1, v. 0, vi. 1
- i. $2 * 3$, ii. 4
- i. 2, ii. 0, iii. 1, iv. 0, v. 0, vi. 0
- i. 2, ii. 10
- My name is George Malkovich
- i. (-3) , ii. 1
- California California

Chapter 8

8.2 Review Questions: True/False

1. false
2. true
3. false
4. false

8.3 Review Exercises

1. Solution

For the input value of 3

Step	Statement	a	b	c	d
1	<code>a = Int32.Parse(Console.ReadLine())</code>	3	?	?	?
2	<code>a = (a + 1) * (a + 1) + 6 / 3 * 2 + 20</code>	40	?	?	?
3	<code>b = a % 13</code>	40	1	?	?
4	<code>c = b % 7</code>	40	1	1	?
5	<code>d = a * b * c</code>	40	1	1	40
6	<code>Console.WriteLine(a + ", " + b + ", " + c + ", " + d)</code>	It displays: 40, 1, 1, 40			

For the input value of 4

Step	Statement	a	b	c	d
1	<code>a = Int32.Parse(Console.ReadLine())</code>	4	?	?	?
2	<code>a = (a + 1) * (a + 1) + 6 / 3 * 2 + 20</code>	49	?	?	?
3	<code>b = a % 13</code>	49	10	?	?
4	<code>c = b % 7</code>	49	10	3	?
5	<code>d = a * b * c</code>	49	10	3	1470
6	<code>Console.WriteLine(a + ", " + b + ", " + c + ", " + d)</code>	It displays: 49, 10, 3, 1470			

For the input value of 1

Step	Statement	a	b	c	d
1	<code>a = Int32.Parse(Console.ReadLine())</code>	1	?	?	?
2	<code>a = (a + 1) * (a + 1) + 6 / 3 * 2 + 20</code>	28	?	?	?
3	<code>b = a % 13</code>	28	2	?	?
4	<code>c = b % 7</code>	28	2	2	?
5	<code>d = a * b * c</code>	28	2	2	112
6	<code>Console.WriteLine(a + ", " + b + ", " + c + ", " + d)</code>	It displays: 28, 2, 2, 112			

2. Solution

For the input values of 8, 4

Step	Statement	a	b	c	d	e
1	<code>a = Double.Parse(Console.ReadLine())</code>	8.0	?	?	?	?
2	<code>b = Double.Parse(Console.ReadLine())</code>	8.0	4.0	?	?	?

3	<code>c = a + b</code>	8.0	4.0	12.0	?	?
4	<code>d = 1 + a / b * c + 2</code>	8.0	4.0	12.0	27.0	?
5	<code>e = c + d</code>	8.0	4.0	12.0	27.0	39.0
6	<code>c += d + e</code>	8.0	4.0	78.0	27.0	39.0
7	<code>e--</code>	8.0	4.0	78.0	27.0	38.0
8	<code>d -= c + d % c</code>	8.0	4.0	78.0	-78.0	38.0
9	<code>Console.WriteLine(c + ", " + d + ", " + e)</code>	It displays: 78, -78, 38				

For the input values of 4, 4

Step	Statement	a	b	c	d	e
1	<code>a = Double.Parse(Console.ReadLine())</code>	4.0	?	?	?	?
2	<code>b = Double.Parse(Console.ReadLine())</code>	4.0	4.0	?	?	?
3	<code>c = a + b</code>	4.0	4.0	8.0	?	?
4	<code>d = 1 + a / b * c + 2</code>	4.0	4.0	8.0	11.0	?
5	<code>e = c + d</code>	4.0	4.0	8.0	11.0	19.0
6	<code>c += d + e</code>	4.0	4.0	38.0	11.0	19.0
7	<code>e--</code>	4.0	4.0	38.0	11.0	18.0
8	<code>d -= c + d % c</code>	4.0	4.0	38.0	-38.0	18.0
9	<code>Console.WriteLine(c + ", " + d + ", " + e)</code>	It displays: 38, -38, 18				

Chapter 9

9.6 Review Exercises

1. Solution

The statement `S = S1 + S3 + SS` is wrong. It must be `S = S1 + S3 + S5`

2. Solution

For the input values of 5, 5

Step	Statement	a	b	c	d	e
1	<code>a = Int32.Parse(Console.ReadLine())</code>	5	?	?	?	?
2	<code>b = Int32.Parse(Console.ReadLine())</code>	5	5	?	?	?
3	<code>c = a + b</code>	5	5	10	?	?
4	<code>d = 5 + a / b * c + 2</code>	5	5	10	17	?
5	<code>e = c - d</code>	5	5	10	17	-7
6	<code>c += d + c</code>	5	5	37	17	-7
7	<code>e--</code>	5	5	37	17	-8
8	<code>d += e + c % b</code>	5	5	37	11	-8
9	<code>Console.WriteLine(c + ", " + d + ", " + e)</code>	It displays: 37, 11, -8				

For the input values of 4, 2

Step	Statement	a	b	c	d	e
1	<code>a = Int32.Parse(Console.ReadLine())</code>	4	?	?	?	?
2	<code>b = Int32.Parse(Console.ReadLine())</code>	4	2	?	?	?
3	<code>c = a + b</code>	4	2	6	?	?
4	<code>d = 5 + a / b * c + 2</code>	4	2	6	19	?
5	<code>e = c - d</code>	4	2	6	19	-13
6	<code>c += d + c</code>	4	2	31	19	-13
7	<code>e--</code>	4	2	31	19	-14
8	<code>d += e + c % b</code>	4	2	31	6	-14
9	<code>Console.WriteLine(c + ", " + d + ", " + e)</code>	It displays: 31, 6, -14				

3. Solution

For the input value of 5

Step	Statement	a	b	c
1	<code>b = Int32.Parse(Console.ReadLine())</code>	?	5	?
2	<code>c = 5</code>	?	5	5
3	<code>c = c * b</code>	?	5	25
4	<code>a = 3 * c % 10</code>	5	5	25

5	<code>Console.WriteLine(a)</code>	It displays: 5
----------	-----------------------------------	----------------

For the input value of 4

Step	Statement	a	b	c
1	<code>b = Int32.Parse(Console.ReadLine())</code>	?	4	?
2	<code>c = 5</code>	?	4	5
3	<code>c = c * b</code>	?	4	20
4	<code>a = 3 * c % 10</code>	0	4	20
5	<code>Console.WriteLine(a)</code>	It displays: 0		

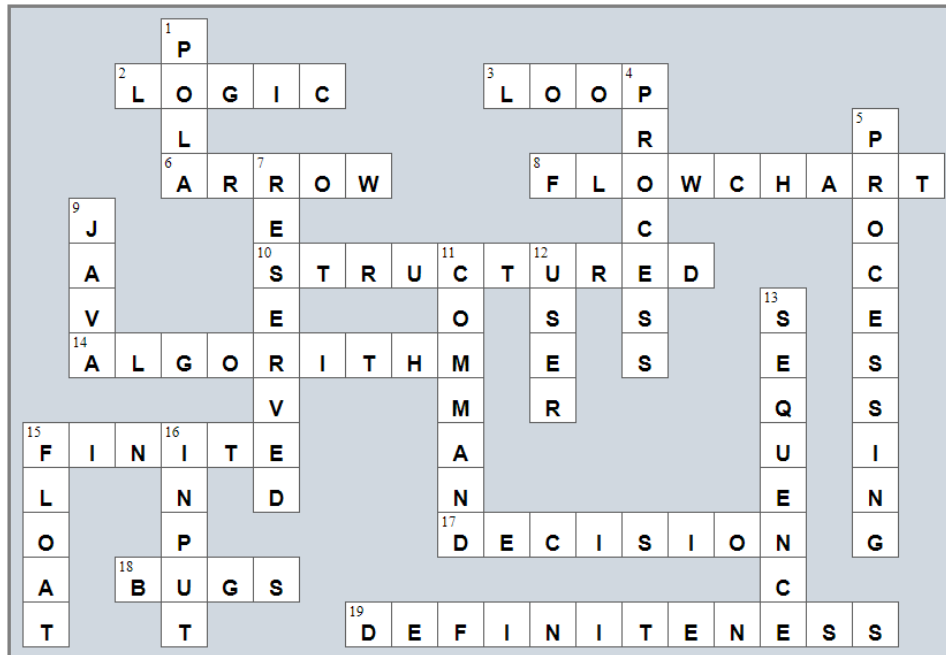
For the input value of 15

Step	Statement	a	b	c
1	<code>b = Int32.Parse(Console.ReadLine())</code>	?	15	?
2	<code>c = 5</code>	?	15	5
3	<code>c = c * b</code>	?	15	75
4	<code>a = 3 * c % 10</code>	5	15	75
5	<code>Console.WriteLine(a)</code>	It displays: 5		

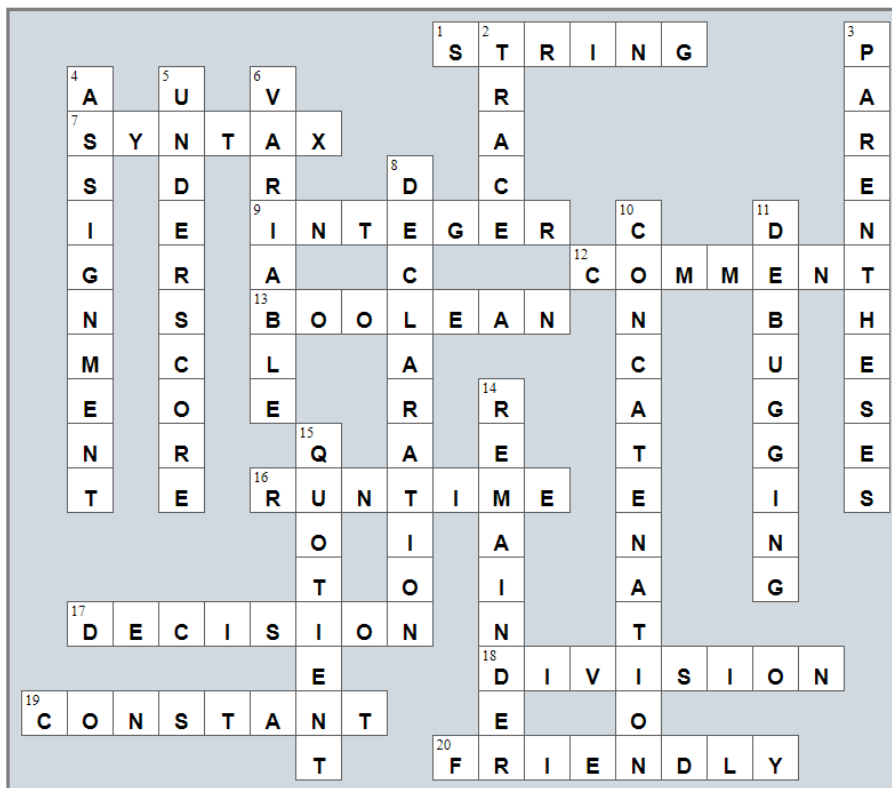
Review in "Getting Started with C#"

Review Crossword Puzzles

1.



2.



Chapter 10

10.2 Review Exercises

1. Solution

```
static void Main(string[] args) {
    double b, h, area;

    Console.Write("Enter base: ");
    b = Double.Parse(Console.ReadLine());
    Console.Write("Enter height: ");
    h = Double.Parse(Console.ReadLine());

    area = 0.5 * b * h;

    Console.WriteLine(area);
}
```

2. Solution

```
static void Main(string[] args) {
    double angle1, angle2, angle3;

    Console.Write("Enter 1st angle: ");
    angle1 = Double.Parse(Console.ReadLine());
    Console.Write("Enter 2nd angle: ");
    angle2 = Double.Parse(Console.ReadLine());

    angle3 = 180 - angle1 - angle2;

    Console.WriteLine(angle3);
}
```

3. Solution

```
static void Main(string[] args) {
    int g1, g2, g3, g4;
    double average;

    Console.Write("Enter 1st grade: ");
    g1 = Double.Parse(Console.ReadLine());
    Console.Write("Enter 2nd grade: ");
    g2 = Double.Parse(Console.ReadLine());
    Console.Write("Enter 3rd grade: ");
    g3 = Double.Parse(Console.ReadLine());
    Console.Write("Enter 4th grade: ");
    g4 = Double.Parse(Console.ReadLine());

    average = (g1 + g2 + g3 + g4) / 4.0;

    Console.WriteLine(average);
}
```

4. Solution

```
const double PI = 3.14159;

static void Main(string[] args) {
    double r, perimeter;

    Console.Write("Enter radius: ");
    r = Double.Parse(Console.ReadLine());

    perimeter = 2 * PI * r;

    Console.WriteLine(perimeter);
}
```

5. Solution

```
const double PI = 3.14159;

static void Main(string[] args) {
    double d, radius, volume;

    Console.Write("Enter diameter (in meters): ");
    d = Double.Parse(Console.ReadLine());

    radius = d / 2;
    volume = 4 / 3 * PI * radius * radius * radius;

    Console.WriteLine(volume);
}
```

6. Solution

Only a), e) and g) are syntactically correct. The latter is more user friendly.

7. Solution

```
const double PI = 3.14159;

static void Main(string[] args) {
    double d, radius, perimeter, area, volume;

    Console.Write("Enter diameter: ");
    d = Double.Parse(Console.ReadLine());

    radius = d / 2;
    perimeter = 2 * PI * radius;
    area = PI * radius * radius;
    volume = 4 / 3 * PI * radius * radius * radius;

    Console.WriteLine(radius + " " + perimeter + " " + area + " " + volume);
}
```

8. Solution

```
static void Main(string[] args) {
    double charge, tip, tax, total;

    Console.Write("Enter charge for a meal: ");
    charge = Double.Parse(Console.ReadLine());

    tip = charge * 10 / 100;
    tax = charge * 7 / 100;

    total = charge + tip + tax;

    Console.WriteLine(total);
}
```

9. Solution

```
static void Main(string[] args) {
    double a, t, s;

    Console.Write("Enter acceleration in m/sec2: ");
    a = Double.Parse(Console.ReadLine());
    Console.Write("Enter time traveled in sec: ");
    t = Double.Parse(Console.ReadLine());

    s = 0.5 * a * t * t;

    Console.WriteLine(s);
}
```

10. Solution

```
static void Main(string[] args) {
    double f, c;

    Console.Write("Enter temperature in Fahrenheit: ");
    f = Double.Parse(Console.ReadLine());

    c = 5 / 9 * (f - 32);

    Console.WriteLine(c);
}
```

11. Solution

```
static void Main(string[] args) {
    int w, h;
    double bmi;

    Console.Write("Enter weight in pounds: ");
    w = Int32.Parse(Console.ReadLine());
    Console.Write("Enter height in inches: ");
    h = Int32.Parse(Console.ReadLine());
}
```

```

    bmi = w * 703.0 / (h * h);

    Console.WriteLine(bmi);
}

```

12. Solution

```

static void Main(string[] args) {
    double s_total, g_rate, tip, total;

    Console.Write("Enter subtotal: ");
    s_total = Double.Parse(Console.ReadLine());
    Console.Write("Enter gratuity rate (0 - 100): ");
    g_rate = Double.Parse(Console.ReadLine());

    tip = s_total * g_rate / 100;

    total = s_total + tip;

    Console.WriteLine("Tip is $" + tip + " and total is $" + total);
}

```

13. Solution

```

const double VAT = 0.20;

static void Main(string[] args) {
    double btax_price1, btax_price2, btax_price3, atax_price1, atax_price2, atax_price3, avg;

    Console.Write("Enter before-tax price 1: ");
    btax_price1 = Double.Parse(Console.ReadLine());
    Console.Write("Enter before-tax price 2: ");
    btax_price2 = Double.Parse(Console.ReadLine());
    Console.Write("Enter before-tax price 3: ");
    btax_price3 = Double.Parse(Console.ReadLine());

    atax_price1 = btax_price1 + btax_price1 * VAT;
    atax_price2 = btax_price2 + btax_price2 * VAT;
    atax_price3 = btax_price3 + btax_price3 * VAT;

    avg = (atax_price1 + atax_price2 + atax_price3) / 3;

    Console.WriteLine(avg);
}

```

14. Solution

```

const int VAT = 0.20;

static void Main(string[] args) {
    double atax_price, btax_price;

    Console.Write("Enter after-tax price: ");
    atax_price = Int32.Parse(Console.ReadLine());
}

```

```

    btax_price = atax_price / (1 + VAT);

    Console.WriteLine(btax_price);
}

```

15. Solution

```

static void Main(string[] args) {
    double i_price, discount, f_price, saved;

    Console.Write("Enter price: ");
    i_price = Double.Parse(Console.ReadLine());
    Console.Write("Enter discount: ");
    discount = Double.Parse(Console.ReadLine());

    f_price = i_price - i_price * discount / 100;
    saved = i_price - f_price;

    Console.WriteLine(f_price + " " + saved);
}

```

16. Solution

```

const int VAT = 0.20;

static void Main(string[] args) {
    int i_kWh, f_kWh, kWh_consumed;
    double cost;

    Console.Write("Enter kWh at the beginning of the month: ");
    i_kWh = Int32.Parse(Console.ReadLine());
    Console.Write("Enter kWh at the end of the month: ");
    f_kWh = Int32.Parse(Console.ReadLine());

    kWh_consumed = f_kWh - i_kWh;

    cost = kWh_consumed * 0.06;
    cost += cost * VAT;

    Console.WriteLine(kWh_consumed + " " + cost);
}

```

17. Solution

```

static void Main(string[] args) {
    int day, month, days_passed, days_left;

    Console.Write("Enter current month: ");
    month = Int32.Parse(Console.ReadLine());
    Console.Write("Enter current day: ");
    day = Int32.Parse(Console.ReadLine());

    days_passed = (month - 1) * 30 + day;
    days_left = 360 - days_passed;
}

```



```
    Console.WriteLine(days_left);  
}
```

Chapter 11

11.3 Review Questions: True/False

- | | | |
|----------|----------|-----------|
| 1. true | 5. false | 9. true |
| 2. false | 6. false | 10. true |
| 3. false | 7. true | 11. false |
| 4. false | 8. true | 12. false |

11.4 Review Questions: Multiple Choice

- | | | |
|------|------|------|
| 1. d | 3. b | 5. a |
| 2. d | 4. c | 6. b |

11.5 Review Exercises

1. Solution

For the input value of 9

Step	Statement	a	b	c
1	a = Double.Parse(Console.ReadLine())	9.0	?	?
2	a += 6 / Math.Sqrt(a) * 2 + 20.4	33.4	?	?
3	b = Math.Round(a) % 4	33.4	1.0	?
4	c = b % 3	33.4	1.0	1.0
5	Console.WriteLine(a + ", " + b + ", " + c)	It displays: 33.4, 1, 1		

For the input value of 4

Step	Statement	a	b	c
1	a = Double.Parse(Console.ReadLine())	4.0	?	?
2	a += 6 / Math.Sqrt(a) * 2 + 20.4	30.4	?	?
3	b = Math.Round(a) % 4	30.4	2.0	?
4	c = b % 3	30.4	2.0	2.0
5	Console.WriteLine(a + ", " + b + ", " + c)	It displays: 30.4, 2, 2		

2. Solution

For the input value of -2

Step	Statement	a	b	c
1	a = Int32.Parse(Console.ReadLine())	-2	?	?
2	b = Math.Abs(a) % 4 + Math.Pow(a, 4)	-2	18	?
3	c = b % 5	-2	18	3
4	Console.WriteLine(b + ", " + c)	It displays: 18, 3		

For the input value of -3

Step	Statement	a	b	c
1	<code>a = Int32.Parse(Console.ReadLine())</code>	-3	?	?
2	<code>b = Math.Abs(a) % 4 + Math.Pow(a, 4)</code>	-3	84	?
3	<code>c = b % 5</code>	-3	84	4
4	<code>Console.WriteLine(b + ", " + c)</code>	It displays: 84, 4		

3. Solution

```
static void Main(string[] args) {
    double degrees, radians;

    Console.Write("Enter angle in radians: ");
    radians = Double.Parse(Console.ReadLine());

    degrees = radians * 180 / Math.PI;

    Console.WriteLine(degrees);
}
```

4. Solution

```
static void Main(string[] args) {
    double a, b, hypotenuse;

    Console.Write("Enter right angle side A of a right-angled triangle: ");
    a = Double.Parse(Console.ReadLine());
    Console.Write("Enter right angle side B of a right-angled triangle: ");
    b = Double.Parse(Console.ReadLine());

    hypotenuse = Math.Sqrt(Math.Pow(a, 2) + Math.Pow(b, 2));

    Console.WriteLine(hypotenuse);
}
```

5. Solution

```
static void Main(string[] args) {
    double adjacent, opposite, th;

    Console.Write("Enter angle  $\theta$  (in degrees) of a right-angled triangle: ");
    th = Double.Parse(Console.ReadLine());
    Console.Write("Enter length of adjacent side: ");
    adjacent = Double.Parse(Console.ReadLine());

    opposite = Math.Tan(th * Math.PI / 180) * adjacent;

    Console.WriteLine(opposite);
}
```

Chapter 12

12.2 Review Exercises

1. Solution

- i. a, e, g, h
- ii. c, f

2. Solution

- i. $y = \text{Math.Pow}(x + 3, 5 * w) / (7 * (x - 4))$
- ii. $y = \text{Math.Pow}(3 * \text{Math.Pow}(x, 2) - \text{Math.Pow}(x, 3) / 4, 1 / 5.0)$
- iii. $y = \text{Math.Sqrt}(\text{Math.Pow}(x, 4) - 2 * \text{Math.Pow}(x, 3) - 7 * x * x + x) / \text{Math.Pow}(4 * (7 * \text{Math.Pow}(x, 4) - 3 / 4.0 * \text{Math.Pow}(x, 3)) * (7 * x * x + x), 1/3.0)$
- iv. $y = x / (x - 3 * (x - 1)) + x * \text{Math.Pow}(x - 1, 1 / 5.0) / ((\text{Math.Pow}(x, 3) - 2) * \text{Math.Pow}(x - 1, 3))$
- v. $y = \text{Math.Pow}(\text{Math.Sin}(\text{Math.PI} / 3) - \text{Math.Cos}(\text{Math.PI} / 2 * w), 2)$
- vi. $y = \text{Math.Pow}(\text{Math.Sin}(\text{Math.PI} / 2 * x) + \text{Math.Cos}(3 * \text{Math.PI} / 2 * w), 3) / \text{Math.Pow}(\text{Math.Tan}(2 * \text{Math.PI} / 3 * w) - \text{Math.Sin}(\text{Math.PI} / 2 * x), 0.5) + 6$

3. Solution

```
static void Main(string[] args) {
    double x, y;

    Console.Write("Enter value for x: ");
    x = Double.Parse(Console.ReadLine());

    y = Math.Sqrt(x) * (Math.Pow(x, 3) + Math.Pow(x, 2));

    Console.WriteLine(y);
}
```

4. Solution

```
static void Main(string[] args) {
    double x, y;

    Console.Write("Enter value for x: ");
    x = Double.Parse(Console.ReadLine());

    y = 7 * x / (2 * x + 4 * (x * x + 4));

    Console.WriteLine(y);
}
```

5. Solution

```
static void Main(string[] args) {
    double w, x, y;
```

```

Console.Write("Enter value for x: ");
x = Double.Parse(Console.ReadLine());
Console.Write("Enter value for w: ");
w = Double.Parse(Console.ReadLine());

y = Math.Pow(x, x + 1) / Math.Pow(Math.Tan(2 * w / 3 + 5) - Math.Tan(x / 2 + 1), 3);

Console.WriteLine(y);
}

```

6. Solution

```

static void Main(string[] args) {
    double w, x, y;

    Console.Write("Enter value for x: ");
    x = Double.Parse(Console.ReadLine());
    Console.Write("Enter value for w: ");
    w = Double.Parse(Console.ReadLine());

    y = (3 + w) / (6 * x - 7 * (x + 4)) + x * Math.Pow(3 * w + 1, 1 / 5) * (5 * x + 4) / ((Math.Pow(x, 3)
+ 3) * Math.Pow(x - 1, 7));

    Console.WriteLine(y);
}

```

7. Solution

```

static void Main(string[] args) {
    double w, x, y;

    Console.Write("Enter value for x: ");
    x = Double.Parse(Console.ReadLine());
    Console.Write("Enter value for w: ");
    w = Double.Parse(Console.ReadLine());

    y = Math.Pow(x, x) / Math.Pow(Math.Sin(2 * w / 3 + 5) - x, 2) + Math.Pow(Math.Sin(3 * x) + w, x + 1)
/ Math.Pow(Math.Sqrt(7 * w), 3 / 2);

    Console.WriteLine(y);
}

```

8. Solution

```

static void Main(string[] args) {
    double a, b, c, area, semi;

    Console.Write("Enter length A: ");
    a = Double.Parse(Console.ReadLine());
    Console.Write("Enter length B: ");
    b = Double.Parse(Console.ReadLine());
    Console.Write("Enter length C: ");
    c = Double.Parse(Console.ReadLine());
}

```

```
semi = (a + b + c) / 2;  
area = Math.Sqrt(semi * (semi - a) * (semi - b) * (semi - c));  
  
Console.WriteLine(area);  
}
```

Chapter 13

13.2 Review Exercises

1. Solution

```
static void Main(string[] args) {
    int last_digit, n, result;

    Console.Write("Enter an integer: ");
    n = Int32.Parse(Console.ReadLine());

    last_digit = n % 10;
    result = last_digit * 8;

    Console.WriteLine(result);
}
```

2. Solution

```
static void Main(string[] args) {
    int digit1, digit2, digit3, digit4, digit5, number, r, reversed_number;

    Console.Write("Enter a five-digit integer: ");
    number = Int32.Parse(Console.ReadLine());

    digit5 = number % 10;
    r = (int)(number / 10);

    digit4 = r % 10;
    r = (int)(r / 10);

    digit3 = r % 10;
    r = (int)(r / 10);

    digit2 = r % 10;
    digit1 = (int)(r / 10);

    reversed_number = digit5 * 10000 + digit4 * 1000 + digit3 * 100 + digit2 * 10 + digit1;
    Console.WriteLine(reversed_number);
}
```

3. Solution

```
static void Main(string[] args) {
    int n, result;

    Console.Write("Enter an integer: ");
    n = Int32.Parse(Console.ReadLine());

    result = n % 2;

    Console.WriteLine(result);
}
```

4. Solution

```
static void Main(string[] args) {
    int n, result;

    Console.Write("Enter an integer: ");
    n = Int32.Parse(Console.ReadLine());

    result = 1 - n % 2;

    Console.WriteLine(result);
}
```

5. Solution

```
static void Main(string[] args) {
    int days, hours, minutes, number, r, seconds, weeks;

    Console.Write("Enter an elapsed time in seconds: ");
    number = Int32.Parse(Console.ReadLine());

    weeks = (int)(number / 604800); // 60 * 60 * 24 * 7 = 604800
    r = number % 604800;

    days = (int)(r / 86400); // 60 * 60 * 24 = 86400
    r = r % 86400;

    hours = (int)(r / 3600);
    r = r % 3600;

    minutes = (int)(r / 60);
    seconds = r % 60;

    Console.Write(weeks + " week(s) " + days + " day(s) " + hours + " hour(s) ");
    Console.WriteLine(minutes + " minute(s) and " + seconds + " second(s)");
}
```

6. Solution

```
static void Main(string[] args) {
    int amount, r, usd1, usd10, usd20, usd5;

    Console.Write("Enter amount of money to withdraw: ");
    amount = Int32.Parse(Console.ReadLine());

    usd20 = (int)(amount / 20);
    r = amount % 20;

    usd10 = (int)(r / 10);
    r = r % 10;

    usd5 = (int)(r / 5);
    usd1 = r % 5;

    Console.Write(usd20 + " note(s) of $20 " + usd10 + " note(s) of $10 ");
}
```



```
    Console.WriteLine(usd5 + " note(s) of $5 and " + usd1 + " note(s) of $1");  
}
```

7. Solution

```
static void Main(string[] args) {  
    int distance, feet, inches, miles, r, steps, yards;  
  
    Console.Write("Enter number of steps: ");  
    steps = Int32.Parse(Console.ReadLine());  
  
    distance = steps * 25;  
  
    miles = (int)(distance / 63360);  
    r = distance % 63360;  
  
    yards = (int)(r / 36);  
    r = r % 36;  
  
    feet = (int)(r / 12);  
    inches = r % 12;  
  
    Console.Write(miles + " mile(s) " + yards + " yard(s) ");  
    Console.WriteLine(feet + " foot/feet and " + inches + " inch(es)");  
}
```

Chapter 14

14.4 Review Questions: True/False

- | | | |
|----------|-----------|-----------|
| 1. true | 7. true | 13. false |
| 2. false | 8. false | 14. true |
| 3. false | 9. true | 15. true |
| 4. true | 10. false | 16. true |
| 5. true | 11. false | |
| 6. false | 12. true | |

14.5 Review Questions: Multiple Choice

1. d
2. b
3. a
4. d
5. b
6. b
7. c
8. a
9. c

14.6 Review Exercises

1. Solution

```
static void Main(string[] args) {
    string first_name, last_name, middle_name, title;

    Console.Write("First name: ");
    first_name = Console.ReadLine();
    Console.Write("Middle name: ");
    middle_name = Console.ReadLine();
    Console.Write("Last name: ");
    last_name = Console.ReadLine();
    Console.Write("Title: ");
    title = Console.ReadLine();

    Console.WriteLine(title + " " + first_name + " " + middle_name + " " + last_name);
    Console.WriteLine(first_name + " " + middle_name + " " + last_name);
    Console.WriteLine(last_name + ", " + first_name);
    Console.WriteLine(last_name + ", " + first_name + " " + middle_name);
    Console.WriteLine(last_name + ", " + first_name + " " + middle_name + ", " + title);
    Console.WriteLine(first_name + " " + last_name);
}
```

2. Solution

```
static void Main(string[] args) {
    string alphabet, random_word;

    Random rnd = new Random();

    alphabet = "abcdefghijklmnopqrstuvwxyz";

    random_word = (" " + alphabet[rnd.Next(0, 26)]).ToUpper() +
        alphabet[rnd.Next(0, 26)] +
        alphabet[rnd.Next(0, 26)] +
        alphabet[rnd.Next(0, 26)] +
        alphabet[rnd.Next(0, 26)];

    Console.WriteLine(random_word);
}
```

3. Solution

```
static void Main(string[] args) {
    string name, x, secret_password;
    Random rnd = new Random();

    Console.Write("Enter name: ");
    name = Console.ReadLine();

    x = name.ToLower().Replace(" ", "");
}
```

```
secret_password = "" + x[rnd.Next(0, x.Length)] +  
                  x[rnd.Next(0, x.Length)] +  
                  x[rnd.Next(0, x.Length)] +  
                  rnd.Next(1000, 10000);  
  
Console.WriteLine(secret_password);  
}
```

4. Solution

First approach

```
static void Main(string[] args) {  
    int number, reversed_number;  
    string s_number, digit1, digit2, digit3;  
  
    Console.Write("Enter a three-digit integer: ");  
    number = Int32.Parse(Console.ReadLine());  
  
    s_number = "" + number;  
  
    digit1 = "" + s_number[0];  
    digit2 = "" + s_number[1];  
    digit3 = "" + s_number[2];  
  
    reversed_number = 100 * Int32.Parse(digit3) + 10 * Int32.Parse(digit2) + Int32.Parse(digit1);  
  
    Console.WriteLine(reversed_number);  
}
```

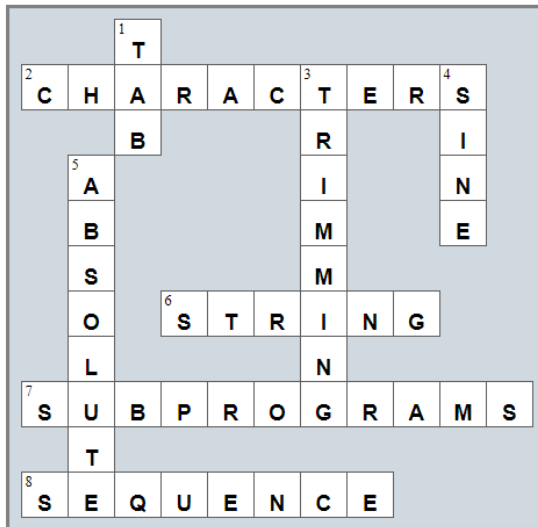
Second approach

```
static void Main(string[] args) {  
    int number, reversed_number;  
    string s_number;  
  
    Console.Write("Enter a three-digit integer: ");  
    number = Int32.Parse(Console.ReadLine());  
  
    s_number = "" + number;  
    reversed_number = Int32.Parse("" + s_number[2] + s_number[1] + s_number[0]);  
  
    Console.WriteLine(reversed_number);  
}
```

Review in "Sequence Control Structures"

Review Crossword Puzzle

1.



Chapter 15

15.9 Review Questions: True/False

- | | | |
|----------|-----------|-----------|
| 1. true | 9. true | 17. false |
| 2. false | 10. true | 18. true |
| 3. false | 11. true | 19. true |
| 4. false | 12. true | 20. false |
| 5. false | 13. true | 21. true |
| 6. false | 14. true | 22. true |
| 7. false | 15. true | 23. true |
| 8. true | 16. false | |

15.10 Review Questions: Multiple Choice

- | | | |
|------|------|------|
| 1. b | 3. a | 5. c |
| 2. a | 4. a | 6. d |

15.11 Review Exercises

1. Solution

- i. c, e, g
- ii. a, j
- iii. d, f
- iv. b, h, i

2. Solution

a	b	c	a != 1	b > a	c / 2 > 2 * a
3	-5	8	true	false	false
1	10	20	false	true	true
-4	-2	-9	true	true	true

3. Solution

Boolean Expression1 (BE1)	Boolean Expression2 (BE2)	BE1 BE2	BE1 && BE2	!(BE2)
false	false	false	false	true
false	true	true	false	false
true	false	true	false	true
true	true	true	true	false

4. Solution

a	b	c	a > 3 c > b && c > 1	a > 3 && c > b c > 1
4	-6	2	true	true

-3	2	-4	false	false
2	5	5	false	true

5. Solution

Expression	Value
<code>Math.Pow(x + y, 3)</code>	8
<code>(x + y) / (Math.Pow(x, 2) - 14)</code>	1
<code>x - 1 == y + 5</code>	true
<code>x > 2 && y == 1</code>	false
<code>x == 1 y == -2 && !(flag == false)</code>	true
<code>!(x >= 3) && (x % 2 > 1)</code>	false

6. Solution

- i. false
- ii. true

7. Solution

- i. `age < 12 && age != 8`
- ii. `age >= 6 && age <= 9 || age == 11`
- iii. `age > 7 && age != 10 && age != 12`
- iv. `age == 6 || age == 9 || age == 11`
- v. `age >= 6 && age <= 12 && age != 8`
- vi. `age != 7 && age != 10`

8. Solution

- i. `x != 4 || y == 3`
- ii. `x + 4 > 0`
- iii. `!(x <= 5) && y != 4`
- iv. `x == false`
- v. `!(x < 4 && z <= 4)`
- vi. `x == 2 || x < -5`

9. Solution

- i. `!(x < 4 || y == 10)`
- ii. `!(x - 2 < 9)`
- iii. `!(!(x < 2) && y == 4)`
- iv. `!(x == false && y != 3)`
- v. First approach: `!(!(x < 2 || y < 2))`
Second approach: `x < 2 || y < 2`
- vi. `!(x == -2 || x > 2)`

Chapter 16

16.2 Review Questions: True/False

- | | |
|----------|----------|
| 1. false | 5. false |
| 2. false | 6. false |
| 3. true | 7. true |
| 4. false | 8. false |

16.3 Review Questions: Multiple Choice

- | | |
|------|------|
| 1. b | 4. d |
| 2. c | 5. d |
| 3. d | |

16.4 Review Exercises

1. Solution

The corrections/additions are in red

```
static void Main(string[] args) {
    double x, y;

    x = Double.Parse(Console.ReadLine());

    y = -5;
    if (x * y / 2 > 20) {
        y *= 2;
        x += 4 * x * x;
    }

    Console.WriteLine(x + " " + y);
}
```

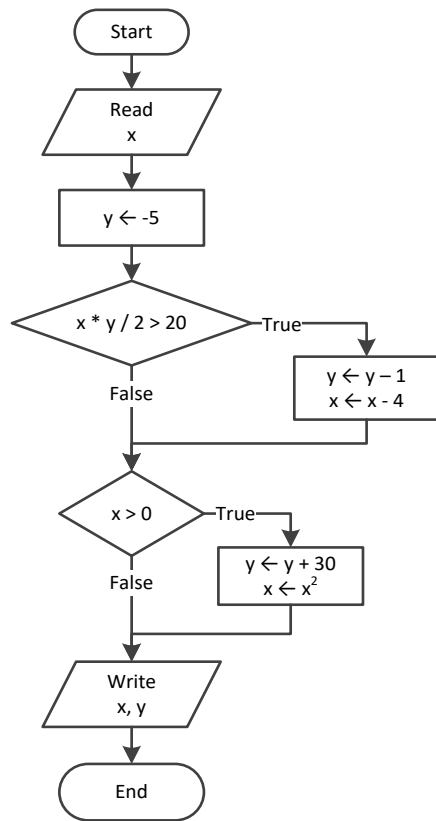
2. Solution

For the input value of 10

Step	Statement	x	y
1	x = Double.Parse(Console.ReadLine())	10.0	?
2	y = -5	10.0	-5.0
3	if (x * y / 2 > 20)	false	
4	if (x > 0)	true	
5	y += 30	10.0	25.0
6	x = Math.Pow(x, 2)	100.0	25.0
7	Console.WriteLine(x + ", " + y)	It displays: 100, 25	

For the input value of -10

Step	Statement	x	y
1	<code>x = Double.Parse(Console.ReadLine())</code>	-10.0	?
2	<code>y = -5</code>	-10.0	-5.0
3	<code>if (x * y / 2 > 20)</code>	true	
4	<code>y--</code>	-10.0	-6.0
5	<code>x -= 4</code>	-14.0	-6.0
6	<code>if (x > 0)</code>	false	
7	<code>Console.WriteLine(x + ", " + y)</code>	It displays: -14, -6	



3. Solution

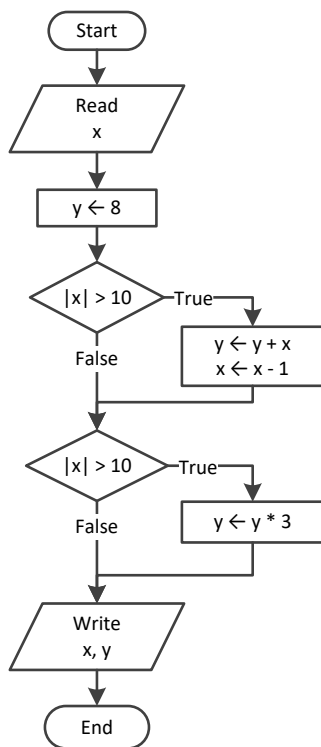
For the input value of -11

Step	Statement	x	y
1	<code>x = Int32.Parse(Console.ReadLine())</code>	-11	?
2	<code>y = 8</code>	-11	8
3	<code>if (Math.Abs(x) > 10)</code>	true	
4	<code>y += x</code>	-11	-3
5	<code>x--</code>	-12	-3
6	<code>if (Math.Abs(x) > 10)</code>	true	
7	<code>y *= 3</code>	-12	-9

8	<code>Console.WriteLine(x + ", " + y)</code>	It displays: -12, -9
----------	--	----------------------

For the input value of 11

Step	Statement	x	y
1	<code>x = Int32.Parse(Console.ReadLine())</code>	11	?
2	<code>y = 8</code>	11	8
3	<code>if (Math.Abs(x) > 10)</code>	true	
4	<code>y += x</code>	11	19
5	<code>x--</code>	10	19
6	<code>if (Math.Abs(x) > 10)</code>	false	
7	<code>Console.WriteLine(x + ", " + y)</code>	It displays: 10, 19	



4. Solution

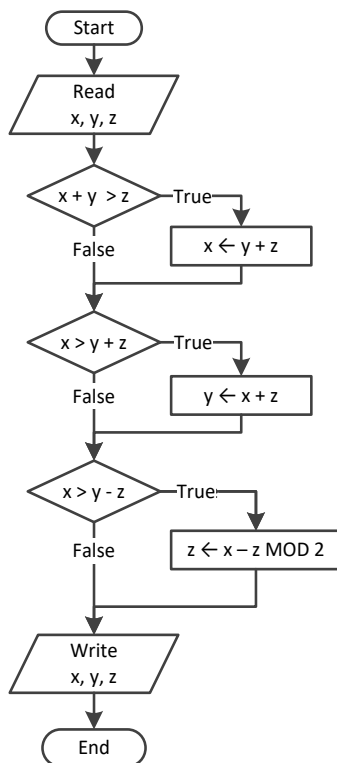
For input values of 1, 2 and 3

Step	Statement	x	y	z
1	<code>x = Int32.Parse(Console.ReadLine())</code>	1	?	?
2	<code>y = Int32.Parse(Console.ReadLine())</code>	1	2	?
3	<code>z = Int32.Parse(Console.ReadLine())</code>	1	2	3
4	<code>if (x + y > z)</code>	false		
5	<code>if (x > y + z)</code>	false		
6	<code>if (x > y - z)</code>	true		
7	<code>z = x - z % 2</code>	1	2	0

8	<code>Console.WriteLine(x + ", " + y + ", " + z)</code>	It displays: 1, 2, 0
----------	---	----------------------

For input values of 4, 2 and 1

Step	Statement	x	y	z
1	<code>x = Int32.Parse(Console.ReadLine())</code>	4	?	?
2	<code>y = Int32.Parse(Console.ReadLine())</code>	4	2	?
3	<code>z = Int32.Parse(Console.ReadLine())</code>	4	2	1
4	<code>if (x + y > z)</code>	true		
5	<code>x = y + z</code>	3	2	1
6	<code>if (x > y + z)</code>	false		
7	<code>if (x > y - z)</code>	true		
8	<code>z = x - z % 2</code>	3	2	2
9	<code>Console.WriteLine(x + ", " + y + ", " + z)</code>	It displays: 3, 2, 2		



5. Solution

```

static void Main(string[] args) {
    double x;

    Console.Write("Enter a number: ");
    x = Double.Parse(Console.ReadLine());

    if (x > 0) {
        Console.WriteLine("Positive");
    }
}

```

6. Solution

```
static void Main(string[] args) {
    double x, y;

    Console.Write("Enter a number: ");
    x = Double.Parse(Console.ReadLine());
    Console.Write("Enter a second number");
    y = Double.Parse(Console.ReadLine());

    if (x > 0 && y > 0) {
        Console.WriteLine("Positive");
    }
}
```

7. Solution

```
static void Main(string[] args) {
    double x, y;

    Console.Write("Enter your age: ");
    x = Double.Parse(Console.ReadLine());

    if (x > 14) {
        Console.WriteLine("You can drive a car in Kansas (USA)");
    }
}
```

8. Solution

```
static void Main(string[] args) {
    string str;

    Console.Write("Enter a string: ");
    str = Console.ReadLine();

    if (str == str.ToUpper()) {
        Console.WriteLine("Uppercase");
    }
}
```

9. Solution

```
static void Main(string[] args) {
    string str;

    Console.Write("Enter a string: ");
    str = Console.ReadLine();

    if (str.Length > 20) {
        Console.WriteLine("Many characters");
    }
}
```

10. Solution

```
static void Main(string[] args) {
    double n1, n2, n3, n4;

    Console.Write("Enter 1st number: ");
    n1 = Double.Parse(Console.ReadLine());
    Console.Write("Enter 2nd number: ");
    n2 = Double.Parse(Console.ReadLine());
    Console.Write("Enter 3rd number: ");
    n3 = Double.Parse(Console.ReadLine());
    Console.Write("Enter 4th number: ");
    n4 = Double.Parse(Console.ReadLine());

    if (n1 < 0 || n2 < 0 || n3 < 0 || n4 < 0) {
        Console.WriteLine("Among the given numbers, there is a negative one!");
    }
}
```

11. Solution

```
static void Main(string[] args) {
    double a, b, c;

    Console.Write("Enter 1st number: ");
    a = Double.Parse(Console.ReadLine());
    Console.Write("Enter 2nd number: ");
    b = Double.Parse(Console.ReadLine());

    if (a > b) {
        c = a;
        a = b;
        b = c;
    }

    Console.WriteLine(a + ", " + b);
}
```

12. Solution

```
static void Main(string[] args) {
    double average, t1, t2, t3;

    Console.Write("Enter 1st temperature: ");
    t1 = Double.Parse(Console.ReadLine());
    Console.Write("Enter 2nd temperature: ");
    t2 = Double.Parse(Console.ReadLine());
    Console.Write("Enter 3rd temperature: ");
    t3 = Double.Parse(Console.ReadLine());

    average = (t1 + t2 + t3) / 3;

    if (average > 60) {
```

```
    Console.WriteLine("Heat Wave");  
  }  
}
```

Chapter 17

17.2 Review Questions: True/False

1. false
2. true
3. true
4. false
5. false
6. false

17.3 Review Questions: Multiple Choice

1. b
2. c
3. c

17.4 Review Exercises

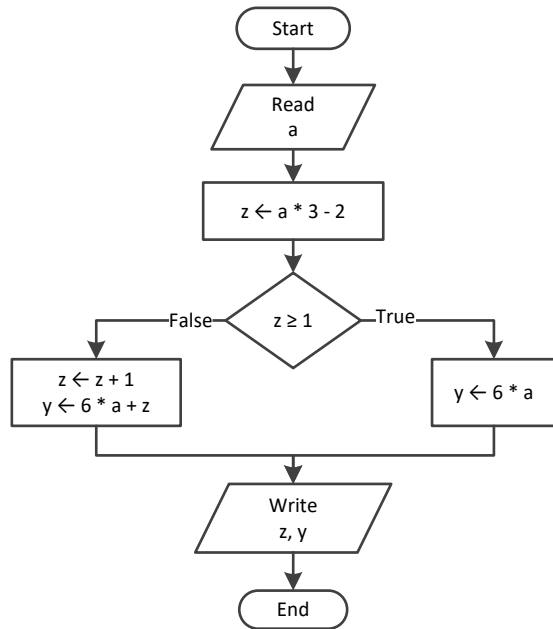
1. Solution

For input value of 3

Step	Statement	a	y	z
1	<code>a = Double.Parse(Console.ReadLine())</code>	3.0	?	?
2	<code>z = a * 3 - 2</code>	3.0	?	7.0
3	<code>if (z >= 1)</code>	true		
4	<code>y = 6 * a</code>	3.0	18.0	7.0
5	<code>Console.WriteLine(z + ", " + y)</code>	It displays: 7 18		

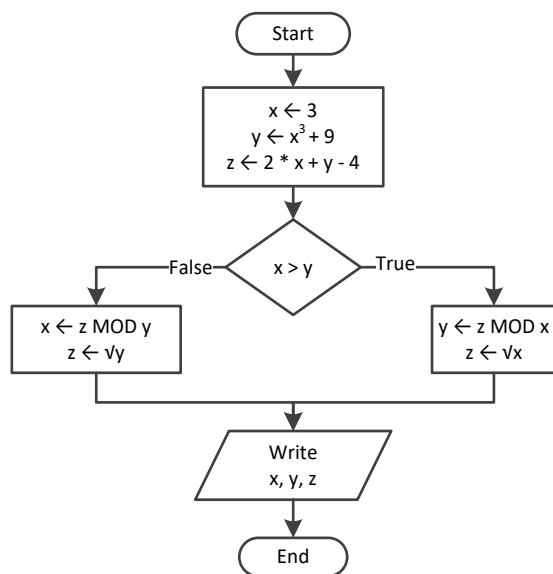
For input value of 0.5

Step	Statement	a	y	z
1	<code>a = Double.Parse(Console.ReadLine())</code>	0.5	?	?
2	<code>z = a * 3 - 2</code>	0.5	?	-0.5
3	<code>if (z >= 1)</code>	false		
4	<code>z++</code>	0.5	?	0.5
5	<code>y = 6 * a + z</code>	0.5	3.5	0.5
6	<code>Console.WriteLine(z + ", " + y)</code>	It displays: 0.5, 3.5		



2. Solution

Step	Statement	x	y	z
1	<code>x = 3</code>	3.0	?	?
2	<code>y = Math.Pow(x, 3) + 9</code>	3.0	36.0	?
3	<code>z = 2 * x + y - 4</code>	3.0	36.0	38.0
4	<code>if (x > y)</code>	false		
5	<code>x = z % y</code>	2.0	36.0	38.0
6	<code>z = Math.Sqrt(y)</code>	2.0	36.0	6.0
7	<code>Console.WriteLine(x + ", " + y + ", " + z)</code>	It displays: 2, 36, 6		



3. Solution

```
static void Main(string[] args) {
    double w, x, y, z;

    x = Double.Parse(Console.ReadLine());
    w = x * 3 - 15;
    z = (w + 7) * (x + 4) - 10;
    if (w > x && z > x) {
        x++;
        y = x / 2 + 4;
    }
    else {
        y = x / 4 + 2;
    }
    Console.WriteLine(y);
}
```

For input value of 10

Step	Statement	x	y	w	z
1	x = Double.Parse(Console.ReadLine())	10.0	?	?	?
2	w = x * 3 - 15	10.0	?	15.0	?
3	z = (w + 7) * (x + 4) - 10	10.0	?	15.0	298.0
4	if (w > x && z > x)	true			
5	x++	11.0	?	15.0	298.0
6	y = x / 2 + 4	11.0	9.5	15.0	298.0
7	Console.WriteLine(y)	It displays: 9.5			

For input value of 2

Step	Statement	x	y	w	z
1	x = Double.Parse(Console.ReadLine())	2.0	?	?	?
2	w = x * 3 - 15	2.0	?	-9.0	?
3	z = (w + 7) * (x + 4) - 10	2.0	?	-9.0	-22.0
4	if (w > x && z > x)	false			
5	y = x / 4 + 2	2.0	2.5	-9.0	-22.0
6	Console.WriteLine(y)	It displays: 2.5			

4. Solution

```
static void Main(string[] args) {
    string name1, name2;
    int goals1, goals2;

    Console.Write("Enter team name 1: ");
    name1 = Console.ReadLine();
    Console.Write("Enter team name 2: ");
    name2 = Console.ReadLine();
}
```

```
Console.Write("Enter goals " + name1 + " scored: ");
goals1 = Int32.Parse(Console.ReadLine());
Console.Write("Enter goals " + name2 + " scored: ");
goals2 = Int32.Parse(Console.ReadLine());

if (goals1 > goals2) {
    Console.WriteLine("Winner: " + name1);
}
else {
    Console.WriteLine("Winner: " + name2);
}
}
```

5. Solution

```
static void Main(string[] args) {
    int x;

    x = Int32.Parse(Console.ReadLine());
    if (x % 6 == 0) {
        Console.WriteLine(x + " is a multiple of 6");
    }
    else {
        Console.WriteLine(x + " is not a multiple of 6");
    }
}
```

6. Solution

```
static void Main(string[] args) {
    int x;

    x = Int32.Parse(Console.ReadLine());
    if (x % 6 == 0 || x % 7 == 0) {
        Console.WriteLine(x + " is a multiple of 6 or a multiple of 7");
    }
    else {
        Console.WriteLine(x + " is neither a multiple of 6 nor a multiple of 7");
    }
}
```

7. Solution

```
static void Main(string[] args) {
    int x, y;

    x = Int32.Parse(Console.ReadLine());

    y = x % 4;
    if (y == 0) {
        Console.WriteLine(x + " is a multiple of 4");
    }
}
```

```

    else {
        Console.WriteLine(x + " is not a multiple of 4");
    }

    Console.WriteLine("The structure is: " + x + " = " + (int)(x / 4) + " x 4 + " + y);
}

```

8. Solution

```

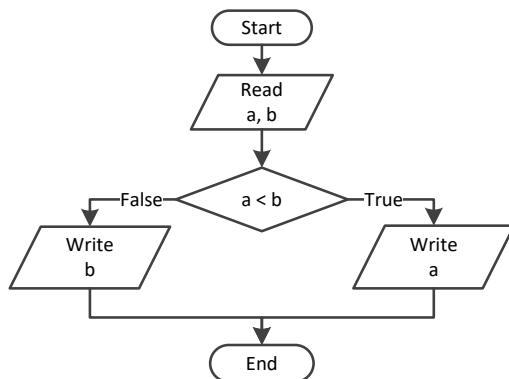
static void Main(string[] args) {
    int x;

    x = Int32.Parse(Console.ReadLine());

    if (x >= 1000 && x <= 9999) {
        Console.WriteLine(x + " is a four-digit integer");
    }
    else {
        Console.WriteLine(x + " is not a four-digit integer");
    }
}

```

9. Solution



```

static void Main(string[] args) {
    double a, b;

    a = Double.Parse(Console.ReadLine());
    b = Double.Parse(Console.ReadLine());

    if (a < b) {
        Console.WriteLine(a);
    }
    else {
        Console.WriteLine(b);
    }
}

```

10. Solution

```

static void Main(string[] args) {
    double a, b, c;

```

```
a = Double.Parse(Console.ReadLine());
b = Double.Parse(Console.ReadLine());
c = Double.Parse(Console.ReadLine());

if (a < b + c && b < a + c && c < a + b) {
    Console.WriteLine("Given numbers can be lengths of the three sides of a triangle");
}
else {
    Console.WriteLine("Given numbers cannot be lengths of the three sides of a triangle");
}
}
```

11. Solution

```
static void Main(string[] args) {
    double a, b, c;

    a = Double.Parse(Console.ReadLine());
    b = Double.Parse(Console.ReadLine());
    c = Double.Parse(Console.ReadLine());

    if (Math.Pow(a, 2) == Math.Pow(b, 2) + Math.Pow(c, 2) ||
        Math.Pow(b, 2) == Math.Pow(a, 2) + Math.Pow(c, 2) ||
        Math.Pow(c, 2) == Math.Pow(a, 2) + Math.Pow(b, 2)) {
        Console.WriteLine("Given numbers can be lengths of the three sides of a right triangle");
    }
    else {
        Console.WriteLine("Given numbers cannot be lengths of the three sides of a right triangle");
    }
}
```

12. Solution

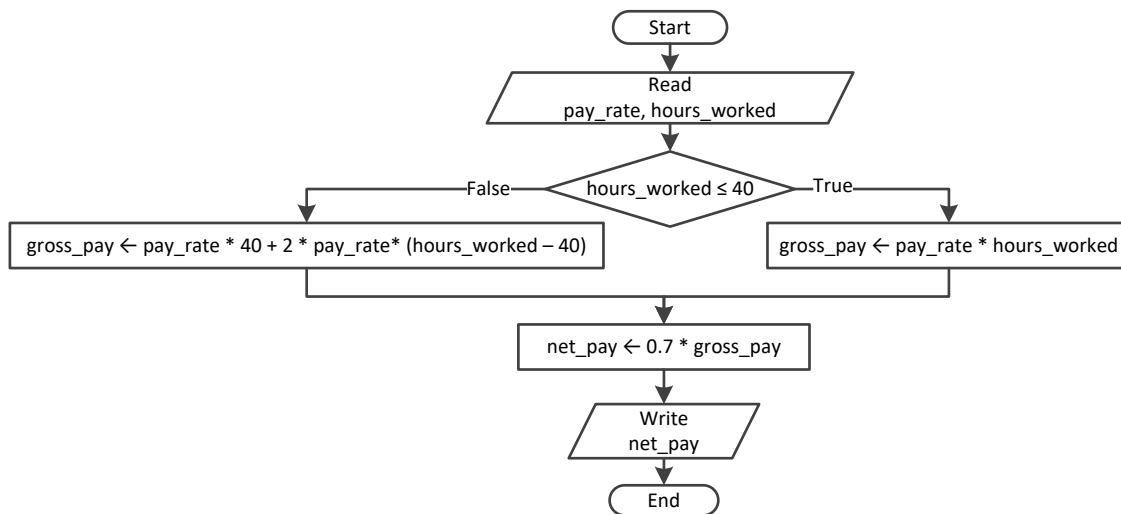
```
static void Main(string[] args) {
    double a, average, b, c;

    Console.Write("Enter 1st jump in meters: ");
    a = Double.Parse(Console.ReadLine());
    Console.Write("Enter 2nd jump in meters: ");
    b = Double.Parse(Console.ReadLine());
    Console.Write("Enter 3rd jump in meters: ");
    c = Double.Parse(Console.ReadLine());

    average = (a + b + c) / 3;

    if (average >= 8) {
        Console.WriteLine("Qualified");
    }
    else {
        Console.WriteLine("Disqualified");
    }
}
```

13. Solution



```

static void Main(string[] args) {
    double gross_pay, net_pay, pay_rate;
    int hours_worked;

    pay_rate = Double.Parse(Console.ReadLine());
    hours_worked = Int32.Parse(Console.ReadLine());

    if (hours_worked <= 40) {
        gross_pay = pay_rate * hours_worked;
    }
    else {
        gross_pay = pay_rate * 40 + 2 * pay_rate * (hours_worked - 40);
    }

    net_pay = 0.7 * gross_pay;
    Console.WriteLine(net_pay);
}

```

14. Solution

```

static void Main(string[] args) {
    int miles, miles_left, r;

    Console.Write("Enter miles traveled: ");
    miles = Int32.Parse(Console.ReadLine());

    r = miles % 12000;

    if (r > 6000) {
        miles_left = 12000 - r;
        Console.WriteLine("Your car needs a major service in " + miles_left + " miles");
    }
    else {
        miles_left = 6000 - r;
        Console.WriteLine("Your car needs a minor service in " + miles_left + " miles");
    }
}

```

```
}  
}
```

15. Solution

```
static void Main(string[] args) {  
    double a1, a2, s1, s2, t;  
  
    Console.Write("Enter the time the two cars traveled: ");  
    t = Double.Parse(Console.ReadLine());  
    Console.Write("Enter the acceleration for car A: ");  
    a1 = Double.Parse(Console.ReadLine());  
    Console.Write("Enter the acceleration for car B: ");  
    a2 = Double.Parse(Console.ReadLine());  
  
    s1 = 0.5 * a1 * Math.Pow(t, 2);  
    s2 = 0.5 * a2 * Math.Pow(t, 2);  
  
    Console.WriteLine("Distance between them: " + Math.Abs(s1 - s2) + " meters");  
  
    if (s1 > s2) {  
        Console.WriteLine("Car A is first");  
    }  
    else {  
        Console.WriteLine("Car B is first");  
    }  
}
```

Chapter 18

18.2 Review Questions: True/False

- | | |
|----------|----------|
| 1. true | 5. false |
| 2. false | 6. true |
| 3. false | 7. false |
| 4. false | 8. true |

18.3 Review Exercises

1. Solution

For input value of 5

Step	Statement	q	b
1	q = Int32.Parse(Console.ReadLine())	5	?
2	if (q > 0 && q <= 50)	true	
3	b = 1	5	1
4	Console.WriteLine(b)	It displays: 1	

For input value of 150

Step	Statement	q	b
1	q = Int32.Parse(Console.ReadLine())	150	?
2	if (q > 0 && q <= 50)	false	
3	else if (q > 50 && q <= 100)	false	
4	else if (q > 100 && q <= 200)	true	
5	b = 3	150	3
6	Console.WriteLine(b)	It displays: 3	

For input value of 250

Step	Statement	q	b
1	q = Int32.Parse(Console.ReadLine())	250	?
2	if (q > 0 && q <= 50)	false	
3	else if (q > 50 && q <= 100)	false	
4	else if (q > 100 && q <= 200)	false	
5	b = 4	250	4
6	Console.WriteLine(b)	It displays: 4	

For input value of -1

Step	Statement	q	b
1	q = Int32.Parse(Console.ReadLine())	-1	?
2	if (q > 0 && q <= 50)	false	
3	else if (q > 50 && q <= 100)	false	
4	else if (q > 100 && q <= 200)	false	

5	b = 4	-1	4
6	Console.WriteLine(b)	It displays: 4	

2. Solution

For input value of 5

Step	Statement	amount	discount	payment
1	amount = Double.Parse(Console.ReadLine())	5.0	?	?
2	discount = 0	5.0	0.0	?
3	if (amount < 20)	true		
4	discount = 0	5.0	0.0	?
5	payment = amount - amount * discount / 100	5.0	0.0	5.0
6	Console.WriteLine(discount + ", " + payment)	It displays: 0, 5		

For input value of 150

Step	Statement	amount	discount	payment
1	amount = Double.Parse(Console.ReadLine())	150.0	?	?
2	discount = 0	150.0	0.0	?
3	if (amount < 20)	false		
4	else if (amount >=20 && amount < 60)	false		
5	else if (amount >= 60 && amount < 100)	false		
6	else if (amount >= 100)	true		
7	discount = 15	150.0	15.0	?
8	payment = amount - amount * discount / 100	150.0	15.0	127.5
9	Console.WriteLine(discount + ", " + payment)	It displays: 15, 127.5		

For input value of -1

Step	Statement	amount	discount	payment
1	amount = Double.Parse(Console.ReadLine())	-1.0	?	?
2	discount = 0	-1.0	0.0	?
3	if (amount < 20)	true		
4	discount = 0	-1.0	0.0	?
5	payment = amount - amount * discount / 100	-1.0	0.0	-1.0
6	Console.WriteLine(discount + ", " + payment)	It displays: 0, -1		

3. Solution

```
static void Main(string[] args) {
    double a, y;

    a = Double.Parse(Console.ReadLine());

    if (a < 1) {
```



```
        y = 5 + a;
        Console.WriteLine(y);
    }
    else if (a < 5) {
        y = 23 / a;
        Console.WriteLine(y);
    }
    else if (a < 10) {
        y = 5 * a;
        Console.WriteLine(y);
    }
    else {
        Console.WriteLine("Error!");
    }
}
```

4. Solution

```
static void Main(string[] args) {
    string name1, name2;
    int goals1, goals2;

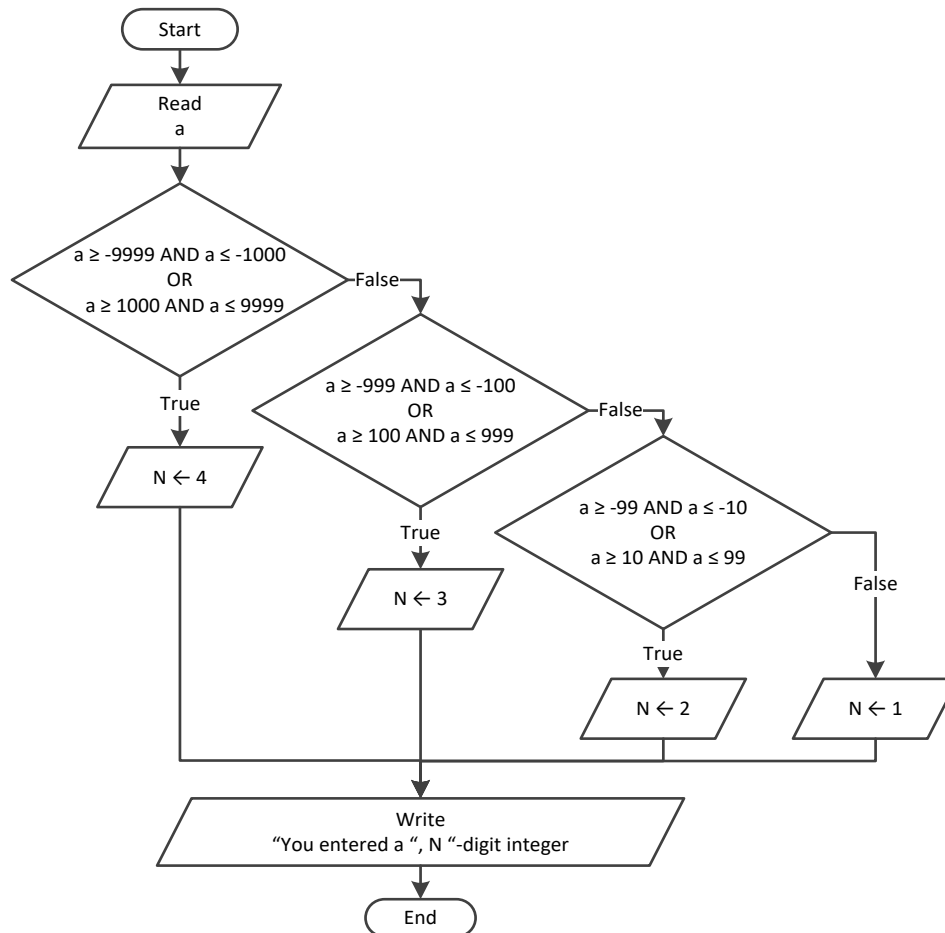
    Console.Write("Enter team name 1: ");
    name1 = Console.ReadLine();
    Console.Write("Enter team name 2: ");
    name2 = Console.ReadLine();

    Console.Write("Enter goals " + name1 + " scored: ");
    goals1 = Int32.Parse(Console.ReadLine());
    Console.Write("Enter goals " + name2 + " scored: ");
    goals2 = Int32.Parse(Console.ReadLine());

    if (goals1 > goals2) {
        Console.WriteLine("Winner: " + name1);
    }
    else if (goals2 > goals1) {
        Console.WriteLine("Winner: " + name2);
    }
    else {
        Console.WriteLine("It's a tie!");
    }
}
```

5. Solution

First approach



```

static void Main(string[] args) {
    int a, n;

    a = Int32.Parse(Console.ReadLine());

    if (a >= -9999 && a <= -1000 || a >= 1000 && a <= 9999) {
        n = 4;
    }
    else if (a >= -999 && a <= -100 || a >= 100 && a <= 999) {
        n = 3;
    }
    else if (a >= -99 && a <= -10 || a >= 10 && a <= 99) {
        n = 2;
    }
    else {
        n = 1;
    }

    Console.WriteLine("You entered a " + n + "-digit integer");
}

```

Second approach

```
static void Main(string[] args) {
    int a;
    string a_string;

    a = Int32.Parse(Console.ReadLine());

    a_string = "" + Math.Abs(a);
    Console.WriteLine("You entered a " + a_string.Length + "-digit integer");
}
```

6. Solution

First approach

```
static void Main(string[] args) {
    int a, n;

    a = Int32.Parse(Console.ReadLine());

    if (a >= -9999 && a <= -1000 || a >= 1000 && a <= 9999) {
        Console.WriteLine("You entered a 4-digit integer");
    }
    else if (a >= -999 && a <= -100 || a >= 100 && a <= 999) {
        Console.WriteLine("You entered a 3-digit integer");
    }
    else if (a >= -99 && a <= -10 || a >= 10 && a <= 99) {
        Console.WriteLine("You entered a 2-digit integer");
    }
    else if (a >= -9 && a <= 9) { //Include the value of zero
        Console.WriteLine("You entered a 1-digit integer");
    }
    else {
        Console.WriteLine("Error: Invalid value!");
    }
}
```

Second approach

```
static void Main(string[] args) {
    int a;
    string a_string;

    a = Int32.Parse(Console.ReadLine());

    if (a >= -9999 && a <= 9999) {
        a_string = "" + Math.Abs(a);
        Console.WriteLine("You entered a " + a_string.Length + "-digit integer");
    }
    else {
        Console.WriteLine("Error: Invalid value!");
    }
}
```

7. Solution

```
static void Main(string[] args) {
    double cad, eur, gbp, jpy, usd;
    int ch;

    Console.WriteLine("1. Convert USD to Euro (EUR)");
    Console.WriteLine("2. Convert USD to British Pound Sterling (GBP)");
    Console.WriteLine("3. Convert USD to Japanese Yen (JPY)");
    Console.WriteLine("4. Convert USD to Canadian Dollar (CAD)");

    Console.Write("Enter a choice: ");
    ch = Int32.Parse(Console.ReadLine());

    Console.Write("Enter an amount in US dollars: ");
    usd = Double.Parse(Console.ReadLine());

    if (ch == 1) {
        eur = usd * 0.87;
        Console.WriteLine("$" + usd + " = " + eur + " EUR");
    }
    else if (ch == 2) {
        gbp = usd * 0.78;
        Console.WriteLine("$" + usd + " = " + gbp + " GBP");
    }
    else if (ch == 3) {
        jpy = usd * 108.55;
        Console.WriteLine("$" + usd + " = " + jpy + " JPY");
    }
    else {
        cad = usd * 1.33;
        Console.WriteLine("$" + usd + " = " + cad + " CAD");
    }
}
```

8. Solution

```
static void Main(string[] args) {
    int m;

    Console.Write("Enter the number of a month between 1 and 12: ");
    m = Int32.Parse(Console.ReadLine());

    if (m <= 2 || m == 12) {
        Console.WriteLine("Winter");
    }
    else if (m <= 5) {
        Console.WriteLine("Spring");
    }
    else if (m <= 8) {
        Console.WriteLine("Summer");
    }
}
```

```
    else {  
        Console.WriteLine("Fall (Autumn)");  
    }  
}
```

9. Solution

```
static void Main(string[] args) {  
    int m;  
  
    Console.Write("Enter the number of a month between 1 and 12: ");  
    m = Int32.Parse(Console.ReadLine());  
  
    if (m < 1 || m > 12) {  
        Console.WriteLine("Error: Invalid value!");  
    }  
    else if (m <= 2 || m == 12) {  
        Console.WriteLine("Winter");  
    }  
    else if (m <= 5) {  
        Console.WriteLine("Spring");  
    }  
    else if (m <= 8) {  
        Console.WriteLine("Summer");  
    }  
    else {  
        Console.WriteLine("Fall (Autumn)");  
    }  
}
```

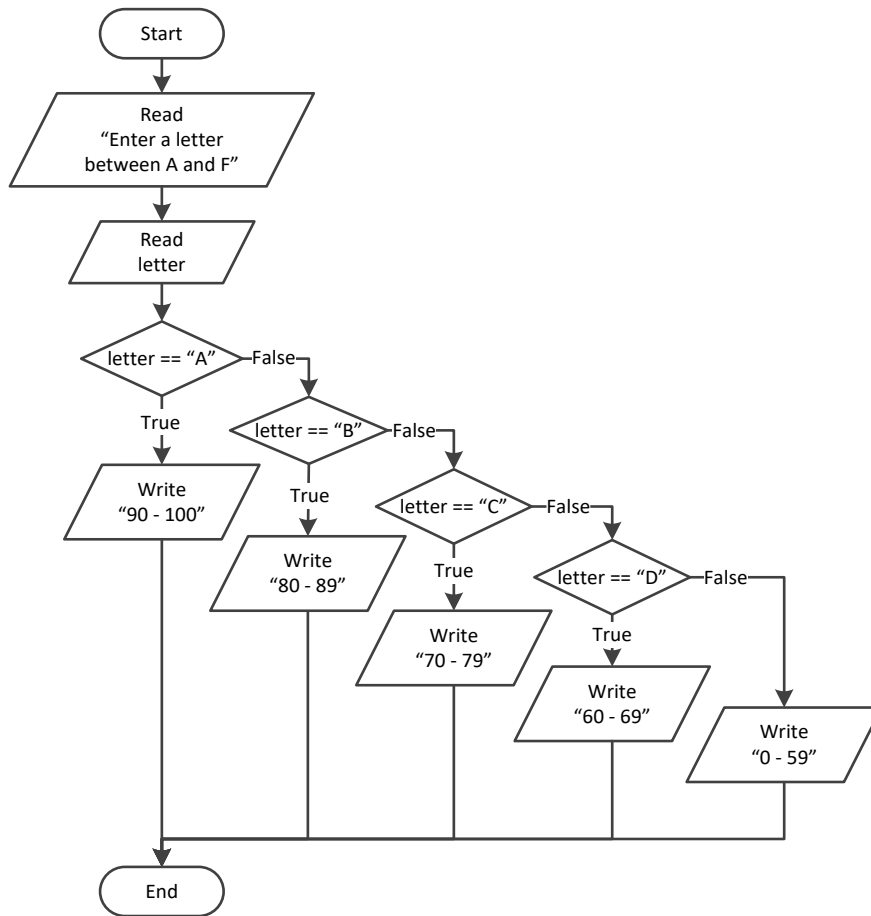
10. Solution

```
static void Main(string[] args) {  
    double n;  
    int x, y;  
  
    Console.Write("Enter a number between 1.0 and 4.9: ");  
    n = Double.Parse(Console.ReadLine());  
  
    x = (int)(n);  
    y = (int)(n * 10) % 10;  
  
    if (x == 1) {  
        Console.Write("One");  
    }  
    else if (x == 2) {  
        Console.Write("Two");  
    }  
    else if (x == 3) {  
        Console.Write("Three");  
    }  
    else if (x == 4) {  
        Console.Write("Four");  
    }  
}
```

```
Console.Write(" point ");

if (y == 1) {
    Console.WriteLine("one");
}
else if (y == 2) {
    Console.WriteLine("two");
}
else if (y == 3) {
    Console.WriteLine("three");
}
else if (y == 4) {
    Console.WriteLine("four");
}
else if (y == 5) {
    Console.WriteLine("five");
}
else if (y == 6) {
    Console.WriteLine("six");
}
else if (y == 7) {
    Console.WriteLine("seven");
}
else if (y == 8) {
    Console.WriteLine("eight");
}
else if (y == 9) {
    Console.WriteLine("nine");
}
else if (y == 0) {
    Console.WriteLine("zero");
}
}
```

11. Solution



```

static void Main(string[] args) {
    string letter;

    Console.Write("Enter a letter between A and F: ");
    letter = Console.ReadLine();

    if (letter == "A") {
        Console.WriteLine("90 - 100");
    }
    else if (letter == "B") {
        Console.WriteLine("80 - 89");
    }
    else if (letter == "C") {
        Console.WriteLine("70 - 79");
    }
    else if (letter == "D") {
        Console.WriteLine("60 - 69");
    }
    else {
        Console.WriteLine("0 - 59");
    }
}

```

Chapter 19

19.2 Review Questions: True/False

1. true
2. false
3. true
4. false
5. true
6. false
7. true

19.3 Review Exercises

1. Solution

For input value of 1

Step	Statement	a	x	y
1	a = Int32.Parse(Console.ReadLine())	1	?	?
2	x = 0	1	0	?
3	y = 0	1	0	0
4	case a == 1	true		
5	x = x + 5	1	5	0
6	y = y + 5	1	5	5
7	Console.WriteLine(x + ", " + y)	It displays: 5, 5		

For input value of 3

Step	Statement	a	x	y
1	a = Int32.Parse(Console.ReadLine())	3	?	?
2	x = 0	3	0	?
3	y = 0	3	0	0
4	case a == 1	false		
5	case a == 2	false		
6	case a == 3	true		
7	x = x - 9	3	-9	0
8	y = y + 3	3	-9	3
9	Console.WriteLine(x + ", " + y)	It displays: -9, 3		

For input value of 250

Step	Statement	a	x	y
1	a = Int32.Parse(Console.ReadLine())	250	?	?
2	x = 0	250	0	?
3	y = 0	250	0	0
4	case a == 1	false		
5	case a == 2	false		
6	case a == 3	false		

7	<code>x = x + 3</code>	250	3	0
8	<code>y++</code>	250	3	1
9	<code>Console.WriteLine(x + ", " + y)</code>	It displays: 3, 1		

2. Solution

For input values of 10, 2, 5

Step	Statement	a	x	y
1	<code>a = Int32.Parse(Console.ReadLine())</code>	10	?	?
2	<code>x = Int32.Parse(Console.ReadLine())</code>	10	2	?
3	<code>y = Double.Parse(Console.ReadLine())</code>	10	2	5.0
4	<code>case a == 10</code>	true		
5	<code>x = x % 2</code>	10	0	5.0
6	<code>y = Math.Pow(y, 2)</code>	10	0	25.0
7	<code>Console.WriteLine(x + ", " + y)</code>	It displays: 0, 25		

For input values of 5, 2, 3

Step	Statement	a	x	y
1	<code>a = Int32.Parse(Console.ReadLine())</code>	5	?	?
2	<code>x = Int32.Parse(Console.ReadLine())</code>	5	2	?
3	<code>y = Double.Parse(Console.ReadLine())</code>	5	2	3.0
4	<code>case a == 10</code>	false		
5	<code>case a == 3</code>	false		
6	<code>case a == 5</code>	true		
7	<code>x = x + 4</code>	5	6	3.0
8	<code>y += 7</code>	5	6	10.0
9	<code>Console.WriteLine(x + ", " + y)</code>	It displays: 6, 10		

For input values of 4, 6, 2

Step	Statement	a	x	y
1	<code>a = Int32.Parse(Console.ReadLine())</code>	4	?	?
2	<code>x = Int32.Parse(Console.ReadLine())</code>	4	6	?
3	<code>y = Double.Parse(Console.ReadLine())</code>	4	6	2.0
4	<code>case a == 10</code>	false		
5	<code>case a == 3</code>	false		
6	<code>case a == 5</code>	false		
7	<code>x -= 3</code>	4	3	2.0
8	<code>y++</code>	4	3	3.0
9	<code>Console.WriteLine(x + ", " + y)</code>	It displays: 3, 3		

3. Solution

```
static void Main(string[] args) {
    string name;

    Console.Write("Enter the name of a month: ");
    name = Console.ReadLine();

    switch (name) {
        case "January":
            Console.WriteLine("1");
            break;
        case "February":
            Console.WriteLine("2");
            break;
        case "March":
            Console.WriteLine("3");
            break;
        case "April":
            Console.WriteLine("4");
            break;
        case "May":
            Console.WriteLine("5");
            break;
        case "June":
            Console.WriteLine("6");
            break;
        case "July":
            Console.WriteLine("7");
            break;
        case "August":
            Console.WriteLine("8");
            break;
        case "September":
            Console.WriteLine("9");
            break;
        case "October":
            Console.WriteLine("10");
            break;
        case "November":
            Console.WriteLine("11");
            break;
        case "December":
            Console.WriteLine("12");
            break;
        default:
            Console.WriteLine("Error");
            break;
    }
}
```

4. Solution

```
static void Main(string[] args) {
    int choice;
    double feet, inches, miles, yards;

    Console.WriteLine("1. Convert Miles to Yards");
    Console.WriteLine("2. Convert Miles to Feet");
    Console.WriteLine("3. Convert Miles to Inches");

    Console.Write("Enter a choice: ");
    choice = Int32.Parse(Console.ReadLine());

    switch (choice) {
        case 1:
            Console.Write("Enter miles: ");
            miles = Double.Parse(Console.ReadLine());
            yards = miles * 1760;
            Console.WriteLine(miles + " miles = " + yards + " yards");
            break;
        case 2:
            Console.Write("Enter miles: ");
            miles = Double.Parse(Console.ReadLine());
            feet = miles * 5280;
            Console.WriteLine(miles + " miles = " + feet + " feet");
            break;
        case 3:
            Console.Write("Enter miles: ");
            miles = Double.Parse(Console.ReadLine());
            inches = miles * 63360;
            Console.WriteLine(miles + " miles = " + inches + " inches");
            break;
        default:
            Console.WriteLine("Invalid choice!");
            break;
    }
}
```

5. Solution

```
static void Main(string[] args) {
    string roman;

    Console.Write("Enter a Roman numeral between I and X: ");
    roman = Console.ReadLine();

    switch (roman) {
        case "I":
            Console.WriteLine("1");
            break;
        case "II":
            Console.WriteLine("2");
```

```
        break;
    case "III":
        Console.WriteLine("3");
        break;
    case "IV":
        Console.WriteLine("4");
        break;
    case "V":
        Console.WriteLine("5");
        break;
    case "VI":
        Console.WriteLine("6");
        break;
    case "VII":
        Console.WriteLine("7");
        break;
    case "VIII":
        Console.WriteLine("8");
        break;
    case "IX":
        Console.WriteLine("9");
        break;
    case "X":
        Console.WriteLine("10");
        break;
    default:
        Console.WriteLine("Error");
        break;
    }
}
```

6. Solution

```
static void Main(string[] args) {
    int total;

    Console.Write("Enter the total number of CDs purchased in a month: ");
    total = Int32.Parse(Console.ReadLine());

    switch (total) {
        case 1:
            Console.WriteLine("You are awarded 3 points");
            break;
        case 2:
            Console.WriteLine("You are awarded 10 points");
            break;
        case 3:
            Console.WriteLine("You are awarded 20 points");
            break;
        default:
            Console.WriteLine("You are awarded 45 points");
            break;
    }
}
```

```
}  
}
```

7. Solution

```
static void Main(string[] args) {  
    int i;  
    string name;  
  
    Random rnd = new Random();  
  
    Console.Write("Enter your name: ");  
    name = Console.ReadLine();  
  
    i = rnd.Next(0, 3);  
  
    switch (i) {  
        case 0:  
            Console.WriteLine("Good morning " + name);  
            break;  
        case 1:  
            Console.WriteLine("Good evening " + name);  
            break;  
        case 2:  
            Console.WriteLine("Good night " + name);  
            break;  
    }  
}
```

8. Solution

```
static void Main(string[] args) {  
    string num;  
  
    num = Console.ReadLine();  
  
    switch (num) {  
        case "zero":  
            Console.WriteLine(0);  
            break;  
        case "one":  
            Console.WriteLine(1);  
            break;  
        case "two":  
            Console.WriteLine(2);  
            break;  
        case "three":  
            Console.WriteLine(3);  
            break;  
        case "four":  
            Console.WriteLine(4);  
            break;  
        case "five":
```

```
        Console.WriteLine(5);
        break;
    case "six":
        Console.WriteLine(6);
        break;
    case "seven":
        Console.WriteLine(7);
        break;
    case "eight":
        Console.WriteLine(8);
        break;
    case "nine":
        Console.WriteLine(9);
        break;
    default:
        Console.WriteLine("I don't know this number!");
        break;
    }
}
```

9. Solution

```
static void Main(string[] args) {
    int b;

    Console.Write("Enter Beaufort number: ");
    b = Int32.Parse(Console.ReadLine());

    switch (b) {
        case 0:
            Console.WriteLine("Calm");
            break;
        case 1:
            Console.WriteLine("Light Air");
            break;
        case 2:
            Console.WriteLine("Light breeze");
            break;
        case 3:
            Console.WriteLine("Gentle breeze");
            break;
        case 4:
            Console.WriteLine("Moderate breeze");
            break;
        case 5:
            Console.WriteLine("Fresh breeze");
            break;
        case 6:
            Console.WriteLine("Strong breeze");
            break;
        case 7:
            Console.WriteLine("Moderate gale");
```

```
        break;
    case 8:
        Console.WriteLine("Gale");
        break;
    case 9:
        Console.WriteLine("Strong gale");
        break;
    case 10:
        Console.WriteLine("Storm");
        break;
    case 11:
        Console.WriteLine("Violent storm");
        break;
    case 12:
        Console.WriteLine("Hurricane force");
        break;
    default:
        Console.WriteLine("Invalid Beaufort number!");
        break;
    }
}
```

Chapter 20

20.3 Review Questions: True/False

1. true
2. true
3. false
4. false
5. true

20.4 Review Exercises

1. Solution

For input values of 20, 1

Step	Statement	x	y
1	<code>x = Int32.Parse(Console.ReadLine())</code>	20	?
2	<code>y = Int32.Parse(Console.ReadLine())</code>	20	1
3	<code>if (x < 30)</code>	true	
4	<code>case y == 1</code>	true	
5	<code>x = x % 3</code>	2	1
6	<code>y = 5</code>	2	5
7	<code>Console.WriteLine(x + ", " + y)</code>	It displays: 2, 5	

For input values of 20, 3

Step	Statement	x	y
1	<code>x = Int32.Parse(Console.ReadLine())</code>	20	?
2	<code>y = Int32.Parse(Console.ReadLine())</code>	20	3
3	<code>if (x < 30)</code>	true	
4	<code>case y == 1</code>	false	
5	<code>case y == 2</code>	false	
6	<code>case y == 3</code>	true	
7	<code>x = x + 5</code>	25	3
8	<code>y += 3</code>	25	6
9	<code>Console.WriteLine(x + ", " + y)</code>	It displays: 25, 6	

For input values of 12, 8

Step	Statement	x	y
1	<code>x = Int32.Parse(Console.ReadLine())</code>	12	?
2	<code>y = Int32.Parse(Console.ReadLine())</code>	12	8
3	<code>if (x < 30)</code>	true	
4	<code>case y == 1</code>	false	
5	<code>case y == 2</code>	false	
6	<code>case y == 3</code>	false	
7	<code>x -= 2</code>	10	8

8	y++	10	9
9	Console.WriteLine(x + ", " + y)	It displays: 10, 9	

For input values of 50, 0

Step	Statement	x	y
1	x = Int32.Parse(Console.ReadLine())	50	?
2	y = Int32.Parse(Console.ReadLine())	50	0
3	if (x < 30)	false	
4	y++	50	1
5	Console.WriteLine(x + ", " + y)	It displays: 50, 1	

2. Solution

For input values of 60, 25

Step	Statement	x	y
1	x = Int32.Parse(Console.ReadLine())	60	?
2	y = Int32.Parse(Console.ReadLine())	60	25
3	if ((x + y) / 2 <= 20)	false	
4	if (y < 15)	false	
5	else if (y < 23)	false	
6	x = 2 * x + 5	125	25
7	y += 1	125	26
8	Console.WriteLine(x + ", " + y)	It displays: 125, 26	

For input values of 50, 8

Step	Statement	x	y
1	x = Int32.Parse(Console.ReadLine())	50	?
2	y = Int32.Parse(Console.ReadLine())	50	8
3	if ((x + y) / 2 <= 20)	false	
4	if (y < 15)	true	
5	x = x % 4	2	8
6	y = 2	2	2
7	Console.WriteLine(x + ", " + y)	It displays: 2, 2	

For input values of 20, 15

Step	Statement	x	y
1	x = Int32.Parse(Console.ReadLine())	20	?
2	y = Int32.Parse(Console.ReadLine())	20	15
3	if ((x + y) / 2 <= 20)	true	
4	if (y < 10)	false	
5	else if (y < 20)	true	

6	<code>x = x * 5</code>	100	15
7	<code>y += 2</code>	100	17
8	<code>Console.WriteLine(x + ", " + y)</code>	It displays: 100, 17	

For input values of 10, 30

Step	Statement	x	y
1	<code>x = Int32.Parse(Console.ReadLine())</code>	10	?
2	<code>y = Int32.Parse(Console.ReadLine())</code>	10	30
3	<code>if ((x + y) / 2 <= 20)</code>	true	
4	<code>if (y < 10)</code>	false	
5	<code>else if (y < 20)</code>	false	
6	<code>x = x - 2</code>	8	30
7	<code>y += 3</code>	8	33
8	<code>Console.WriteLine(x + ", " + y)</code>	It displays: 8, 33	

3. Solution

```
static void Main(string[] args) {
    int a;

    a = Int32.Parse(Console.ReadLine());

    if (a > 1000)
        Console.WriteLine("Big Positive");
    else {
        if (a > 0)
            Console.WriteLine("Positive");
        else {
            if (a < -1000)
                Console.WriteLine("Big Negative");
            else {
                if (a < 0)
                    Console.WriteLine("Negative");
                else
                    Console.WriteLine("Zero");
            }
        }
    }
}
```

4. Solution

```
static void Main(string[] args) {
    double a, b, c;

    Console.Write("Enter the three sides of a triangle: ");
    a = Double.Parse(Console.ReadLine());
    b = Double.Parse(Console.ReadLine());
```

```

c = Double.Parse(Console.ReadLine());

if (a >= b + c || b >= a + c || c >= a + b) {
    Console.WriteLine("Given numbers cannot be lengths of the three sides of a triangle");
}
else {
    if (a == b && b == c) {
        Console.WriteLine("Equilateral");
    }
    else if (Math.Pow(a, 2) == Math.Pow(b, 2) + Math.Pow(c, 2) ||
             Math.Pow(b, 2) == Math.Pow(a, 2) + Math.Pow(c, 2) ||
             Math.Pow(c, 2) == Math.Pow(a, 2) + Math.Pow(b, 2)) {

        Console.WriteLine("Right (or right-angled)");
    }
    else {
        Console.WriteLine("Not special");
    }
}
}

```

5. Solution

```

static void Main(string[] args) {
    int amount, pin, r, usd1, usd10, usd5;

    Console.Write("Enter your four-digit PIN : ");
    pin = Int32.Parse(Console.ReadLine());
    if (pin != 1234) {
        Console.Write("Wrong PIN. Enter your four-digit PIN : ");
        pin = Int32.Parse(Console.ReadLine());
        if (pin != 1234) {
            Console.Write("Wrong PIN. Enter your four-digit PIN : ");
            pin = Int32.Parse(Console.ReadLine());
        }
    }

    if (pin != 1234) {
        Console.WriteLine("PIN locked!");
    }
    else {
        Console.Write("Enter the amount of money (an integer value) that you want to withdraw: ");
        amount = Int32.Parse(Console.ReadLine());
        usd10 = (int)(amount / 10);
        r = amount % 10;
        usd5 = (int)(r / 5);
        usd1 = r % 5;
        Console.Write(usd10 + " note(s) of $10 " + usd5 + " note(s) of $5 ");
        Console.WriteLine("and " + usd1 + " note(s) of $1");
    }
}

```

6. Solution

First approach

```
static void Main(string[] args) {
    double t, w;

    Console.Write("Enter temperature (in Fahrenheit): ");
    t = Double.Parse(Console.ReadLine());
    Console.Write("Enter wind speed (in miles/hour): ");
    w = Double.Parse(Console.ReadLine());

    if (t > 75) {
        if (w > 12) {
            Console.WriteLine("The day is hot and windy");
        }
        else {
            Console.WriteLine("The day is hot and not windy");
        }
    }
    else {
        if (w > 12) {
            Console.WriteLine("The day is cold and windy");
        }
        else {
            Console.WriteLine("The day is cold and not windy");
        }
    }
}
```

Second approach

```
static void Main(string[] args) {
    double t, w;
    string message1, message2;

    Console.Write("Enter temperature (in Fahrenheit): ");
    t = Double.Parse(Console.ReadLine());
    Console.Write("Enter wind speed (in miles/hour): ");
    w = Double.Parse(Console.ReadLine());

    if (t > 75) {
        message1 = "hot";
    }
    else {
        message1 = "cold";
    }

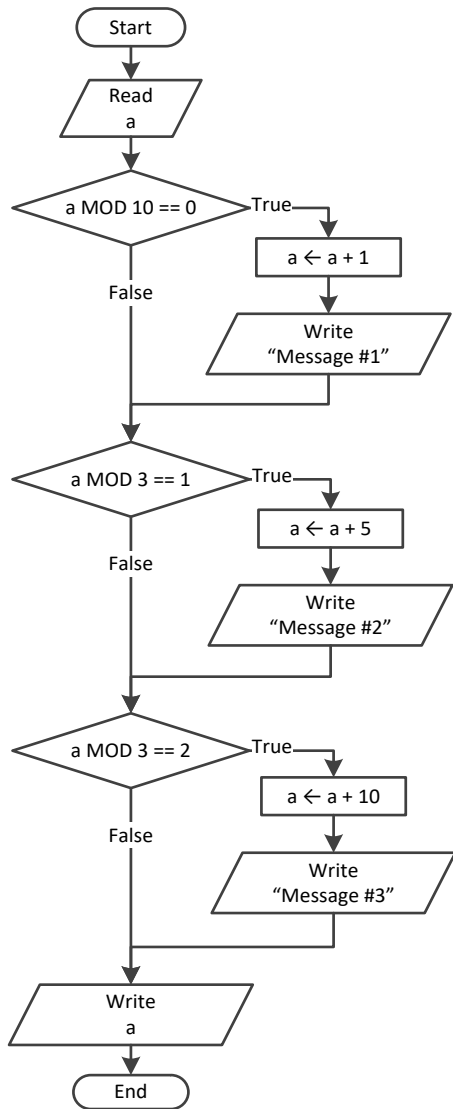
    if (w > 12) {
        message2 = "windy";
    }
    else {
        message2 = "not windy";
    }
}
```

```
    Console.WriteLine("The day is " + message1 + " and " + message2);  
}
```

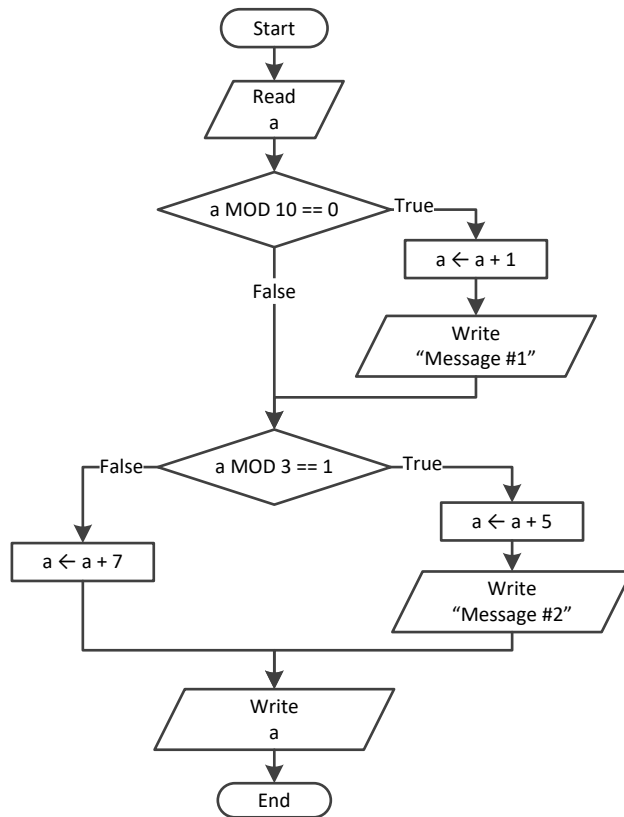
Chapter 21

21.4 Review Exercises

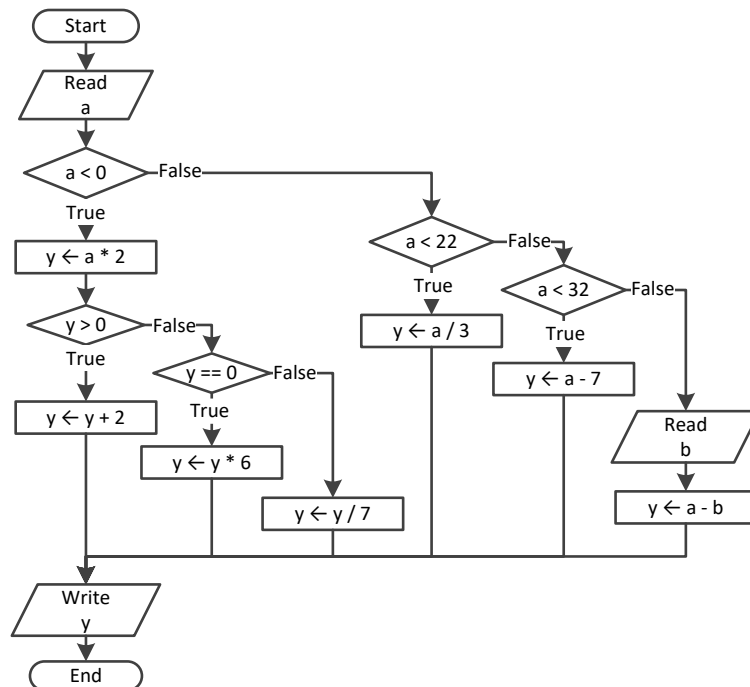
1. Solution



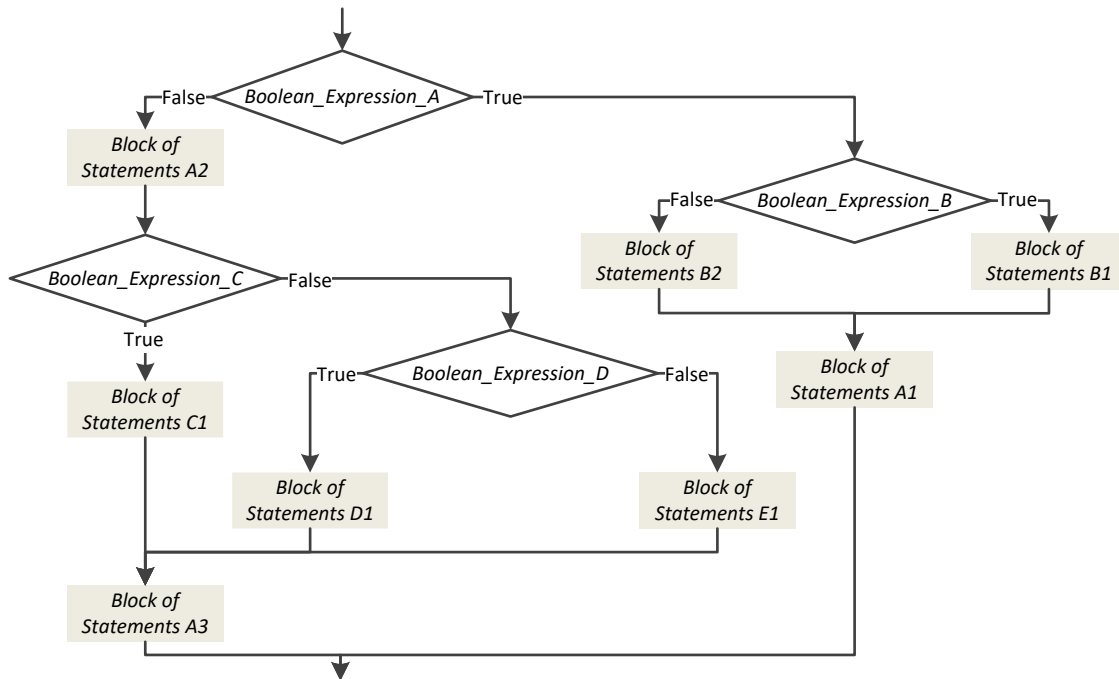
2. Solution



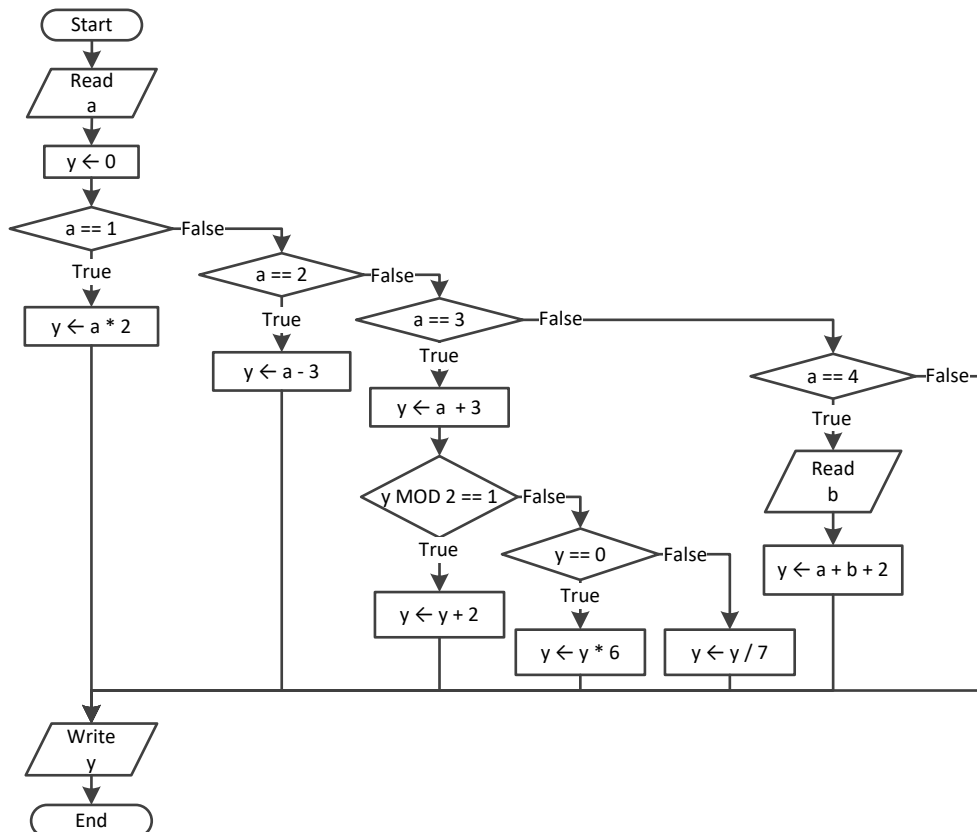
3. Solution



4. Solution



5. Solution



6. Solution

```
static void Main(string[] args) {
    double x, y, z;

    x = Double.Parse(Console.ReadLine());
    y = Double.Parse(Console.ReadLine());

    if (x != 100 || y <= 10) {
        z = Double.Parse(Console.ReadLine());
        if (z <= x + y) {
            x -= 3;
            y = x + 4;
        }
    }
    Console.WriteLine(x + " " + y);
}
```

7. Solution

```
static void Main(string[] args) {
    int x;

    x = Int32.Parse(Console.ReadLine());

    if (x == 1) {
        Console.WriteLine("Good Morning");
        Console.WriteLine("How do you do?");
        Console.WriteLine("Is everything okay?");
    }
    else if (x == 2) {
        Console.WriteLine("Good Evening");
        Console.WriteLine("How do you do?");
        Console.WriteLine("Is everything okay?");
    }
    else if (x == 3) {
        Console.WriteLine("Good Afternoon");
        Console.WriteLine("Is everything okay?");
    }
    else {
        Console.WriteLine("Good Night");
    }
}
```

8. Solution

```
static void Main(string[] args) {
    int a, b, c, d, y;

    a = Int32.Parse(Console.ReadLine());
    b = Int32.Parse(Console.ReadLine());

    c = a % 2;
```

```
d = (int) (b / 5);

if (a >= b)
    y = 1;
else if (d > c && a > 2)
    y = 2;
else if (d * c > a / b) {
    if (d * c > 10)
        y = 4;
    else
        y = 3;
}
else
    y = 5;

Console.WriteLine(y);
}
```

9. Solution

```
static void Main(string[] args) {
    int x;

    x = Int32.Parse(Console.ReadLine());

    if (x > 0) {
        if (x % 10 == 0) {
            Console.WriteLine("Last digit equal to 0");
        }
        else if (x % 10 == 1) {
            Console.WriteLine("Last digit equal to 1");
        }
        else {
            Console.WriteLine("None");
        }
    }
    else {
        if (x == -1) {
            Console.WriteLine("Bye");
        }
        else {
            Console.WriteLine("Invalid Number");
        }
    }
}
```

10. Solution

```
static void Main(string[] args) {
    double a, b, y;

    a = Double.Parse(Console.ReadLine());
    b = Double.Parse(Console.ReadLine());
```

```
y = a * b;

if (y > 0) {
    y--;
    y /= 2;
}
else {
    y +=10;
    if (y > 0) {
        y /= 2;
    }
    else {
        y *= 2;
    }
}
}
```

11. Solution

```
static void Main(string[] args) {
    double a, b, c;

    a = Double.Parse(Console.ReadLine());
    b = Double.Parse(Console.ReadLine());
    c = Double.Parse(Console.ReadLine());

    c = a * b + c;
    if (c > 0) {
        c /= 2;
        if (a > b) {
            a *= 2;
            b *= 2;
        }
        else {
            c /= 20;
            if (c <= 10) {
                b *= 2;
            }
        }
    }
    else {
        c /= 3;
        c /= 20;
        if (c <= 10) {
            b *= 2;
        }
    }
    Console.WriteLine(a + " " + b + " " + c);
}
```

Chapter 22

22.9 Review Questions: True/False

- | | |
|----------|----------|
| 1. false | 5. true |
| 2. true | 6. false |
| 3. false | 7. false |
| 4. true | |

22.10 Review Questions: Multiple Choice

1. a
2. b
3. a
4. c

22.11 Review Exercises

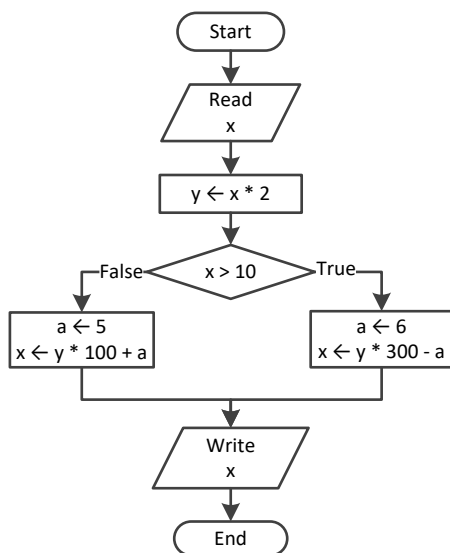
1. Solution

```
static void Main(string[] args) {
    int a, x, y;

    y = Int32.Parse(Console.ReadLine());
    x = Int32.Parse(Console.ReadLine());

    if (y > 0) {
        a = x * 4 * y + 1;
    }
    else {
        a = x * 2 * y + 6;
    }
    Console.WriteLine(y);
    Console.WriteLine(a);
}
```

2. Solution



3. Solution

```
static void Main(string[] args) {
    double a, y;

    a = Double.Parse(Console.ReadLine());

    if (a >= 10) {
        Console.WriteLine("Error!");
    }
    else {
        if (a < 1) {
            y = 5 + a;
        }
    }
}
```

```

    }
    else if (a < 5) {
        y = 23 / a;
    }
    else {
        y = 5 * a;
    }
    Console.WriteLine(y);
}
}

```

4. Solution

```

static void Main(string[] args) {
    int day, month;
    string name;

    day = Int32.Parse(Console.ReadLine());
    month = Int32.Parse(Console.ReadLine());
    name = Console.ReadLine();

    if (day == 16 && month == 2 && name == "Loukia") {
        Console.WriteLine("Happy Birthday!!!");
    }
    else {
        Console.WriteLine("No match!");
    }
}

```

5. Solution

It does not operate the same way when variable *a* is less than or equal to 10. The correct program is

```

static void Main(string[] args) {
    double a, b, c, d;

    a = Double.Parse(Console.ReadLine());
    b = Double.Parse(Console.ReadLine());
    c = Double.Parse(Console.ReadLine());

    if (a > 10) {
        if (c < 2000) {
            d = (a + b + c) / 12;
            Console.WriteLine("The result is: " + d);
        }
        else {
            Console.WriteLine("Error!");
        }
    }
    else {
        Console.WriteLine("Error!");
    }
}

```

6. Solution

```
static void Main(string[] args) {
    double a, b, c, d;

    a = Double.Parse(Console.ReadLine());
    b = Double.Parse(Console.ReadLine());
    c = Double.Parse(Console.ReadLine());

    if (a > 10 && b < 2000 && c != 10) {
        d = (a + b + c) / 12;
        Console.WriteLine("The result is: " + d);
    }

    if (a <= 10) {
        Console.WriteLine("Error!");
    }
}
```

7. Solution

```
static void Main(string[] args) {
    int a, b, y;

    a = Int32.Parse(Console.ReadLine());
    b = Int32.Parse(Console.ReadLine());

    y = 3;
    if (a > 0) {
        y = y * a;
        Console.WriteLine("Hello Zeus");
    }

    Console.WriteLine(y + " " + b);
}
```

8. Solution

```
static void Main(string[] args) {
    double a, b, y;

    a = Double.Parse(Console.ReadLine());
    b = Double.Parse(Console.ReadLine());

    y = 0;
    if (a > 0) {
        y = y + 7;
    }
    else {
        Console.WriteLine("Hello Zeus");
        Console.WriteLine(Math.Abs(a));
    }
    Console.WriteLine(y);
}
```

```
}
```

9. Solution

```
static void Main(string[] args) {  
    string os;  
  
    Console.Write("What is your tablet's OS? ");  
    os = Console.ReadLine();  
  
    if (os == "iOS") {  
        Console.WriteLine("Apple");  
    }  
    else if (os == "Android") {  
        Console.WriteLine("Google");  
    }  
    else if (os == "Windows") {  
        Console.WriteLine("Microsoft");  
    }  
}
```

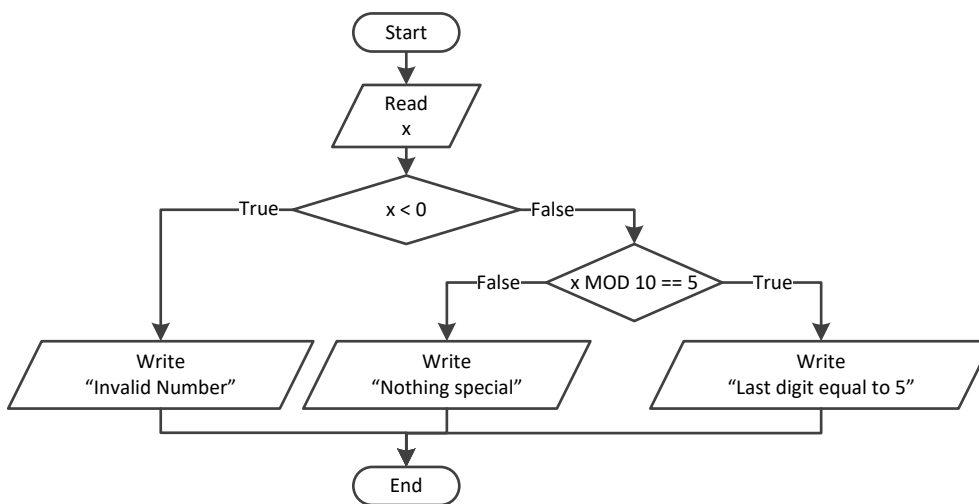

Chapter 23

23.6 Review Exercises

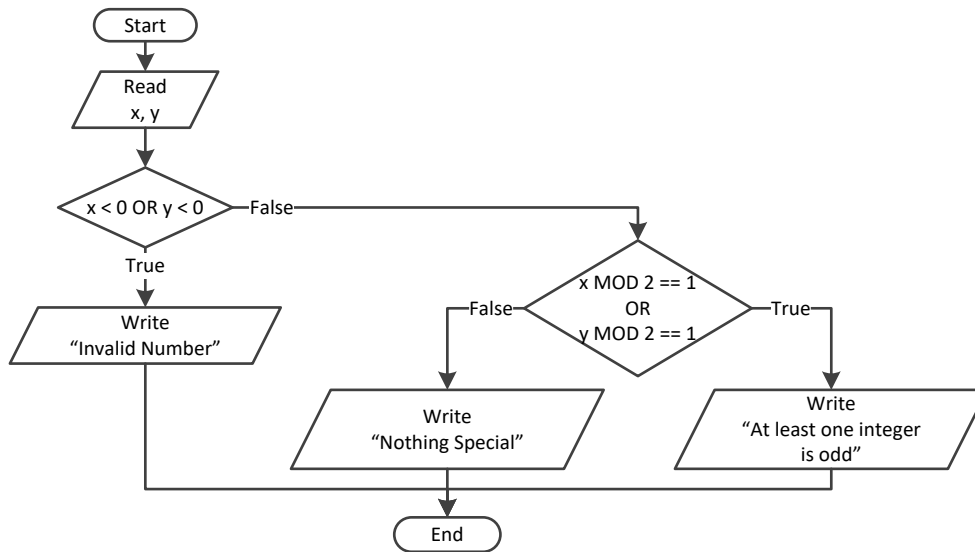
1. Solution

```
static void Main(string[] args) {  
    double x;  
  
    Console.Write("Enter a non-negative number: ");  
    x = Double.Parse(Console.ReadLine());  
    if (x < 0) {  
        Console.WriteLine("Error! You entered a negative value");  
    }  
    else {  
        Console.WriteLine("The square root of " + x + " is " + Math.Sqrt(x));  
    }  
}
```

2. Solution



3. Solution



```

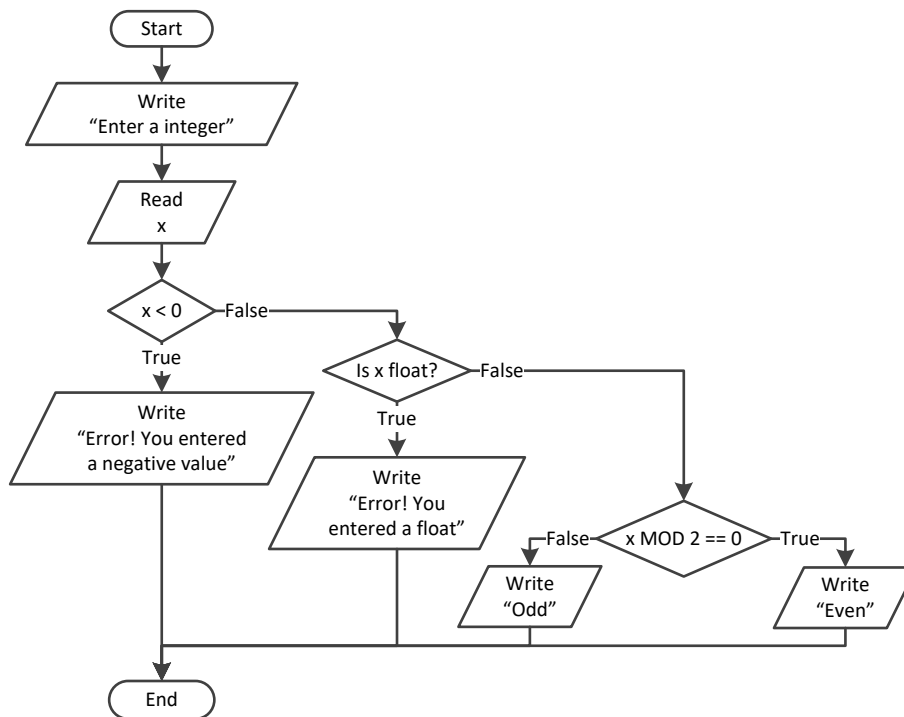
static void Main(string[] args) {
    int x, y;

    x = Int32.Parse(Console.ReadLine());
    y = Int32.Parse(Console.ReadLine());

    if (x < 0 || y < 0) {
        Console.WriteLine("Invalid Number");
    }
    else {
        if (x % 2 == 1 || y % 2 == 1) {
            Console.WriteLine("At least one integer is odd");
        }
        else {
            Console.WriteLine("Nothing Special");
        }
    }
}

```

4. Solution



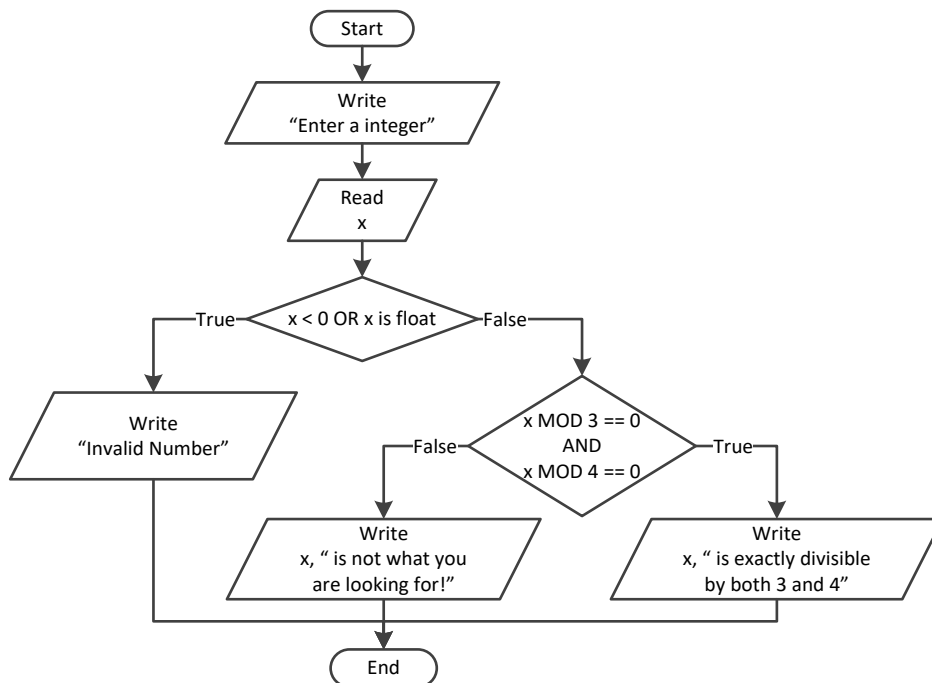
```

static void Main(string[] args) {
    double x;

    Console.Write("Enter a non-negative number: ");
    x = Double.Parse(Console.ReadLine());
    if (x < 0) {
        Console.WriteLine("Error! You entered a negative value");
    }
    else if (x != (int)x) {
        Console.WriteLine("Error! You entered a float");
    }
    else if (x % 2 == 0) {
        Console.WriteLine("Even");
    }
    else {
        Console.WriteLine("Odd");
    }
}

```

5. Solution



```

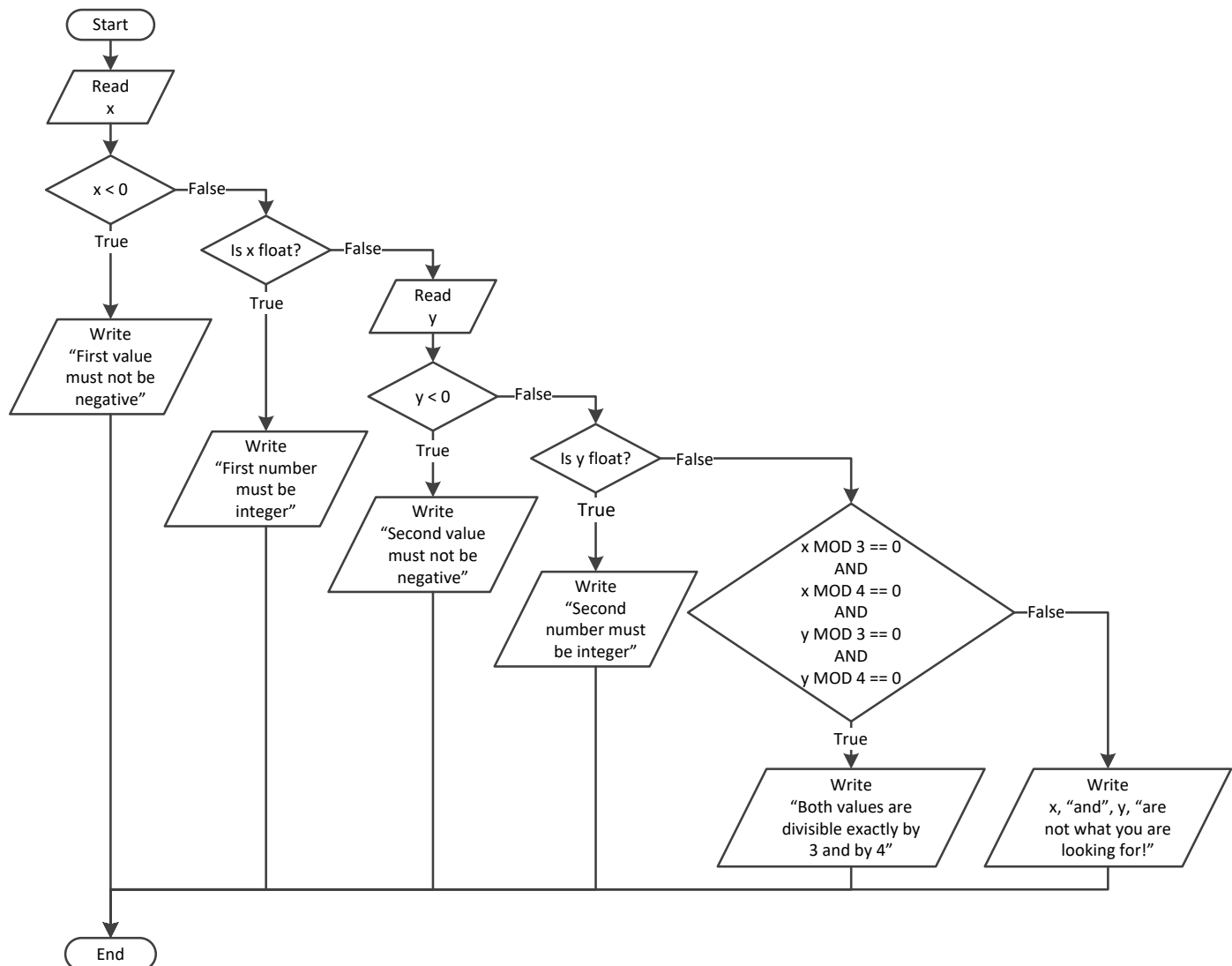
static void Main(string[] args) {
    double x;

    Console.Write("Enter an integer: ");
    x = Double.Parse(Console.ReadLine());

    if (x < 0 || x != (int)x) {
        Console.WriteLine("Invalid Number");
    }
    else if (x % 3 == 0 && x % 4 == 0) {
        Console.WriteLine(x + " is exactly divisible by both 3 and 4");
    }
    else {
        Console.WriteLine(x + " is not what you are looking for!");
    }
}

```

6. Solution



```

static void Main(string[] args) {
    double x, y;

    x = Double.Parse(Console.ReadLine());

    if (x < 0) {
        Console.WriteLine("First value must be not be negative");
    }
    else {
        if (x != (int)x) {
            Console.WriteLine("First number must be integer");
        }
        else {
            y = Double.Parse(Console.ReadLine());
            if (y < 0) {
                Console.WriteLine("Second value must be not be negative");
            }
        }
    }
}
  
```

```

    else {
        if (y != (int)y) {
            Console.WriteLine("Second number must be integer");
        }
        else {
            if (x % 3 == 0 && x % 4 == 0 && y % 3 == 0 && y % 4 == 0 ) {
                Console.WriteLine("Both values are divisible exactly by 3 and by 4");
            }
            else {
                Console.WriteLine("Nothing Special");
            }
        }
    }
}
}
}
}

```

7. Solution

```

static void Main(string[] args) {
    int choice;
    double t;

    Console.WriteLine("1. Convert Kelvin to Fahrenheit");
    Console.WriteLine("2. Convert Fahrenheit to Kelvin");
    Console.WriteLine("3. Convert Fahrenheit to Celsius");
    Console.WriteLine("4. Convert Celsius to Fahrenheit");

    Console.Write("Enter a choice: ");
    choice = Int32.Parse(Console.ReadLine());
    Console.Write("Enter a temperature: ");
    t = Double.Parse(Console.ReadLine());

    if (choice < 1 || choice > 4) {
        Console.WriteLine("Wrong choice");
    }
    else {
        switch (choice) {
            case 1:
                if (t < 0) { //Absolute zero in Kelvin
                    Console.WriteLine("Wrong temperature");
                }
                else {
                    Console.WriteLine(1.8 * t - 459.67);
                }
                break;
            case 2:
                if (t < -459.67) { //Absolute zero in Fahrenheit
                    Console.WriteLine("Wrong temperature");
                }
                else {
                    Console.WriteLine((t + 459.57) / 1.8);
                }
            }
        }
    }
}

```

```

    }
    break;
case 3:
    if (t < -459.67) { //Absolute zero in Fahrenheit
        Console.WriteLine("Wrong temperature");
    }
    else {
        Console.WriteLine(5 / 9 * (t - 32));
    }
    break;
case 4:
    if (t < -273.15) { //Absolute zero in Celcius
        Console.WriteLine("Wrong temperature");
    }
    else {
        Console.WriteLine(9 / 5 * t + 32);
    }
    break;
}
}
}

```

8. Solution

```

static void Main(string[] args) {
    int a, b;
    string op;

    Console.Write("Enter 1st integer: ");
    a = Int32.Parse(Console.ReadLine());
    Console.Write("Enter type of operation: ");
    op = Console.ReadLine();
    Console.Write("Enter 2nd integer: ");
    b = Int32.Parse(Console.ReadLine());

    switch (op) {
        case "+":
            Console.WriteLine(a + b);
            break;
        case "-":
            Console.WriteLine(a - b);
            break;
        case "*":
            Console.WriteLine(a * b);
            break;
        case "/":
            if (b == 0) {
                Console.WriteLine("Error: Division by zero");
            }
            else {
                Console.WriteLine(a / (double)b);
            }
    }
}

```

```
        break;
    case "DIV":
        if (b == 0) {
            Console.WriteLine("Error: Division by zero");
        }
        else {
            Console.WriteLine((int) (a / b));
        }
        break;
    case "MOD":
        if (b == 0) {
            Console.WriteLine("Error: Division by zero");
        }
        else {
            Console.WriteLine(a % b);
        }
        break;
    case "POWER":
        Console.WriteLine(Math.Pow(a, b));
        break;
    }
}
```

9. Solution

```
static void Main(string[] args) {
    int a, b;
    string op;

    Console.Write("Enter 1st integer: ");
    a = Int32.Parse(Console.ReadLine());
    Console.Write("Enter type of operation: ");
    op = Console.ReadLine();
    Console.Write("Enter 2nd integer: ");
    b = Int32.Parse(Console.ReadLine());

    switch (op) {
        case "+":
            Console.WriteLine(a + b);
            break;
        case "-":
            Console.WriteLine(a - b);
            break;
        case "*":
            Console.WriteLine(a * b);
            break;
        case "/":
            if (b == 0) {
                Console.WriteLine("Error: Division by zero");
            }
            else {
                Console.WriteLine(a / (double)b);
            }
        }
    }
}
```

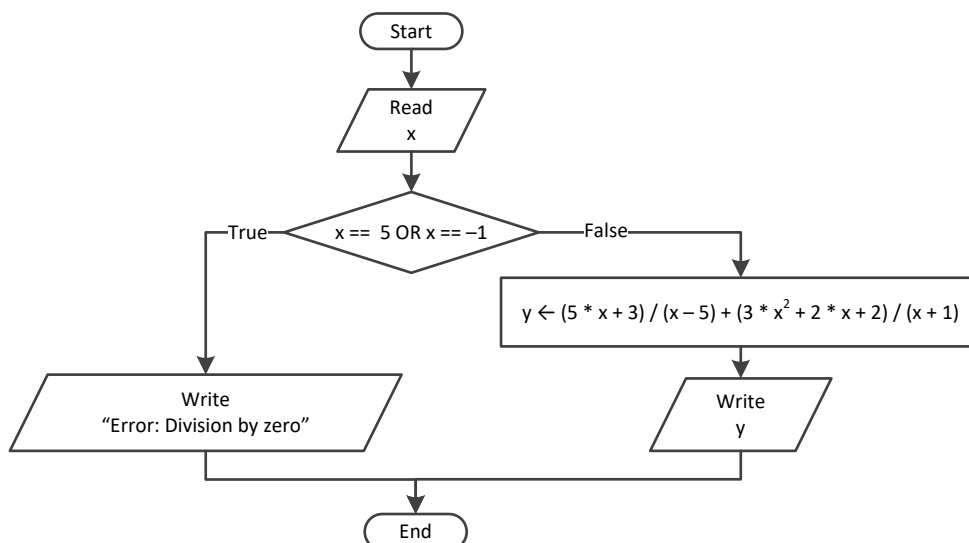


```

    }
    break;
case "DIV":
    if (b == 0) {
        Console.WriteLine("Error: Division by zero");
    }
    else {
        Console.WriteLine((int) (a / b));
    }
    break;
case "MOD":
    if (b == 0) {
        Console.WriteLine("Error: Division by zero");
    }
    else {
        Console.WriteLine(a % b);
    }
    break;
case "POWER":
    Console.WriteLine(Math.Pow(a, b));
    break;
default:
    Console.WriteLine("Error: Invalid operator");
    break;
}
}

```

10. Solution



```

static void Main(string[] args) {
    double x, y;

    x = Double.Parse(Console.ReadLine());

    if (x == 5 || x == -1) {
        Console.WriteLine("Error: Division by zero");
    }
}

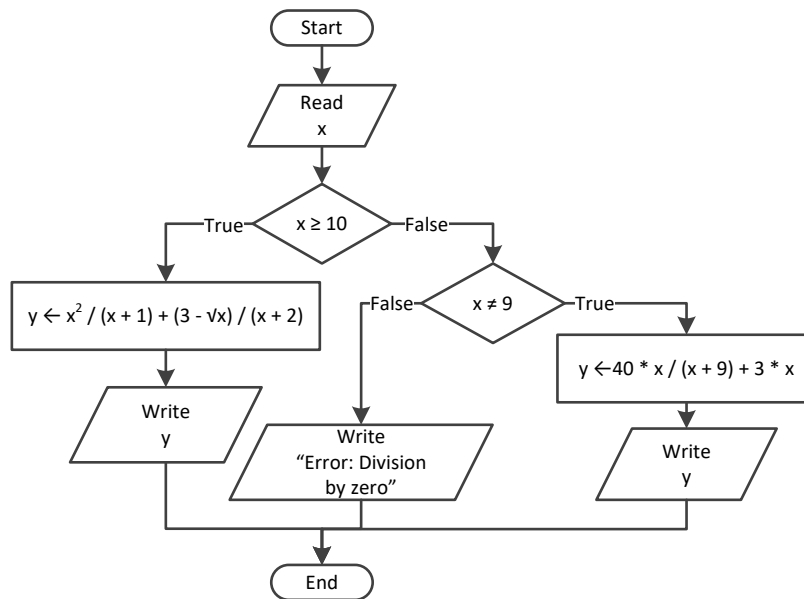
```

```

    }
    else {
        y = (5 * x + 3) / (x - 5) + (3 * Math.Pow(x, 2) + 2 * x + 2) / (x + 1);
        Console.WriteLine(y);
    }
}

```

11. Solution



```

static void Main(string[] args) {
    double x, y;

    x = Double.Parse(Console.ReadLine());
    if (x >= 10) {
        y = Math.Pow(x, 2) / (x + 1) + (3 - Math.Sqrt(x)) / (x + 2);
        Console.WriteLine(y);
    }
    else if (x != 9) {
        y = 40 * x / (x + 9) + 3 * x;
        Console.WriteLine(y);
    }
    else {
        Console.WriteLine("Error: Division by zero");
    }
}

```

12. Solution

```

static void Main(string[] args) {
    double x, y;

    x = Double.Parse(Console.ReadLine());
    if (x <= -15 || x > 25) {
        y = x - 1;
    }
}

```

```

        Console.WriteLine(y);
    }
    else if (x <= -10) {
        y = x / Math.Sqrt(x + 30) + Math.Pow(8 + x, 2) / (x + 1);
        Console.WriteLine(y);
    }
    else if (x <= 0) {
        y = Math.Abs(40 * x) / (x - 8);
        Console.WriteLine(y);
    }
    else {
        if (x == 9) {
            Console.WriteLine("Error: Division by zero");
        }
        else if (x < 9) {
            Console.WriteLine("Error: Invalid square root");
        }
        else {
            y = 3 * x / Math.Sqrt(x - 9);
            Console.WriteLine(y);
        }
    }
}

```

13. Solution

```

static void Main(string[] args) {
    int a1, a2, a3, maximum, minimum;
    string max_name, min_name, n1, n2, n3;

    Console.Write("Enter the age of the first person: ");
    a1 = Int32.Parse(Console.ReadLine());
    Console.Write("Enter the name of the first person: ");
    n1 = Console.ReadLine();
    Console.Write("Enter the age of the second person: ");
    a2 = Int32.Parse(Console.ReadLine());
    Console.Write("Enter the name of the second person: ");
    n2 = Console.ReadLine();
    Console.Write("Enter the age of the third person: ");
    a3 = Int32.Parse(Console.ReadLine());
    Console.Write("Enter the name of the third person: ");
    n3 = Console.ReadLine();

    minimum = a1;
    min_name = n1;
    if (a2 < minimum) {
        minimum = a2;
        min_name = n2;
    }
    if (a3 < minimum) {
        minimum = a3;
        min_name = n3;
    }
}

```

```

    }

    maximum = a1;
    max_name = n1;
    if (a2 > maximum) {
        maximum = a2;
        max_name = n2;
    }
    if (a3 > maximum) {
        maximum = a3;
        max_name = n3;
    }

    Console.WriteLine(min_name + " " + max_name);
}

```

14. Solution

```

static void Main(string[] args) {
    int age1, age2, age3, maximum, middle, minimum;

    Console.Write("Enter age for person No1:");
    age1 = Int32.Parse(Console.ReadLine());
    Console.Write("Enter age for person No2:");
    age2 = Int32.Parse(Console.ReadLine());
    Console.Write("Enter age for person No3:");
    age3 = Int32.Parse(Console.ReadLine());

    minimum = age1;
    if (age2 < minimum) {
        minimum = age2;
    }
    if (age3 < minimum) {
        minimum = age3;
    }

    maximum = age1;
    if (age2 > maximum) {
        maximum = age2;
    }
    if (age3 > maximum) {
        maximum = age3;
    }

    middle = age1 + age2 + age3 - minimum - maximum;
    Console.WriteLine(middle);
}

```

15. Solution

```

static void Main(string[] args) {
    int a1, a2, a3, maximum, minimum, middle;
    string max_name, min_name, n1, n2, n3;

```

```

Console.Write("Enter the age of the first person: ");
a1 = Int32.Parse(Console.ReadLine());
Console.Write("Enter the name of the first person: ");
n1 = Console.ReadLine();
Console.Write("Enter the age of the second person: ");
a2 = Int32.Parse(Console.ReadLine());
Console.Write("Enter the name of the second person: ");
n2 = Console.ReadLine();
Console.Write("Enter the age of the third person: ");
a3 = Int32.Parse(Console.ReadLine());
Console.Write("Enter the name of the third person: ");
n3 = Console.ReadLine();

minimum = a1;
min_name = n1;
if (a2 < minimum) {
    minimum = a2;
    min_name = n2;
}
if (a3 < minimum) {
    minimum = a3;
    min_name = n3;
}

maximum = a1;
max_name = n1;
if (a2 > maximum) {
    maximum = a2;
    max_name = n2;
}
if (a3 > maximum) {
    maximum = a3;
    max_name = n3;
}

middle = a1 + a2 + a3 - minimum - maximum;

if (Math.Abs(maximum - middle) < Math.Abs(minimum - middle)) {
    Console.WriteLine(max_name);
}
else {
    Console.WriteLine(min_name);
}
}

```

16. Solution

```

static void Main(string[] args) {
    int digit1, digit2, digit3, r, total;
    double x;

    Console.Write("Enter a three-digit integer: ");

```

```

x = Double.Parse(Console.ReadLine());

if (x != (int)x) {
    Console.WriteLine("Error! You must enter an integer");
}
else if (x < 100 || x > 999) {
    Console.WriteLine("Entered integer is not a three-digit integer");
}
else {
    digit1 = (int)(x / 100);
    r = (int)x % 100;

    digit2 = (int)(r / 10);
    digit3 = r % 10;

    total = (int)(Math.Pow(digit1, 3) + Math.Pow(digit2, 3) + Math.Pow(digit3, 3));

    if (total == x) {
        Console.WriteLine("You entered an Armstrong number!");
    }
    else {
        Console.WriteLine("You entered a non-Armstrong number!");
    }
}
}

```

17. Solution

```

static void Main(string[] args) {
    int d, m, y;

    Console.Write("Enter day 1 - 31: ");
    d = Int32.Parse(Console.ReadLine());
    Console.Write("Enter month 1 - 12: ");
    m = Int32.Parse(Console.ReadLine());
    Console.Write("Enter year: ");
    y = Int32.Parse(Console.ReadLine());

    if (m == 2) {
        if (y % 4 == 0 && y % 100 != 0 || y % 400 == 0) {
            Console.WriteLine(29 - d);
        }
        else {
            Console.WriteLine(28 - d);
        }
    }
    else if (m == 4 || m == 6 || m == 9 || m == 11) {
        Console.WriteLine(30 - d);
    }
    else {
        Console.WriteLine(31 - d);
    }
}

```

18. Solution

First approach

```
static void Main(string[] args) {
    string word, word1, word2;

    word = Console.ReadLine();

    word1 = word.Substring(0, 1).ToUpper() +
            word.Substring(1, 1).ToLower() +
            word.Substring(2, 1).ToUpper() +
            word.Substring(3, 1).ToLower() +
            word.Substring(4, 1).ToUpper() +
            word.Substring(5, 1).ToLower();

    word2 = word.Substring(0, 1).ToLower() +
            word.Substring(1, 1).ToUpper() +
            word.Substring(2, 1).ToLower() +
            word.Substring(3, 1).ToUpper() +
            word.Substring(4, 1).ToLower() +
            word.Substring(5, 1).ToUpper();

    if (word == word1 || word == word2) {
        Console.WriteLine("Word is okay!");
    }
    else {
        Console.WriteLine("Word is not okay");
    }
}
```

Second approach

```
static void Main(string[] args) {
    string word, word1, word2;

    word = Console.ReadLine();

    word1 = word[0].ToString().ToUpper() +
            word[1].ToString().ToLower() +
            word[2].ToString().ToUpper() +
            word[3].ToString().ToLower() +
            word[4].ToString().ToUpper() +
            word[5].ToString().ToLower();

    word2 = word[0].ToString().ToLower() +
            word[1].ToString().ToUpper() +
            word[2].ToString().ToLower() +
            word[3].ToString().ToUpper() +
            word[4].ToString().ToLower() +
            word[5].ToString().ToUpper();

    if (word == word1 || word == word2) {
        Console.WriteLine("Word is okay!");
    }
}
```

```
    else {  
        Console.WriteLine("Word is not okay");  
    }  
}
```

19. Solution

```
static void Main(string[] args) {  
    int q;  
    double discount, payment;  
  
    Console.Write("Enter quantity: ");  
    q = Int32.Parse(Console.ReadLine());  
  
    if (q < 3) {  
        discount = 0;  
    }  
    else if (q < 6) {  
        discount = 10;  
    }  
    else if (q < 10) {  
        discount = 15;  
    }  
    else if (q < 14) {  
        discount = 20;  
    }  
    else if (q < 20) {  
        discount = 27;  
    }  
    else {  
        discount = 30;  
    }  
  
    payment = q * 10 - q * 10 * discount / 100.0;  
  
    Console.WriteLine("You got a discount of " + discount + "%");  
    Console.WriteLine("You must pay $" + payment);  
}
```

20. Solution

```
const double VAT = 0.19;  
  
static void Main(string[] args) {  
    double amount, discount, payment;  
  
    Console.Write("Enter a before-tax amount: : ");  
    amount = Double.Parse(Console.ReadLine());  
  
    if (amount < 0) {  
        Console.WriteLine("Error! You entered a negative value");  
    }  
    else {
```



```
    if (amount < 50) {
        discount = 0;
    }
    else if (amount < 100) {
        discount = 1;
    }
    else if (amount < 250) {
        discount = 2;
    }
    else {
        discount = 3;
    }

    amount = amount - amount * discount / 100;
    payment = amount + amount * VAT;

    Console.WriteLine("You got a discount of " + discount + "%");
    Console.WriteLine("You must pay $" + payment);
}
}
```

21. Solution

```
static void Main(string[] args) {
    int a, h, w;
    double bmi;

    Console.Write("Enter age: ");
    a = Int32.Parse(Console.ReadLine());
    if (a < 18) {
        Console.WriteLine("Invalid age");
    }
    else {
        Console.Write("Enter weight in pounds: ");
        w = Int32.Parse(Console.ReadLine());
        Console.Write("Enter height in inches: ");
        h = Int32.Parse(Console.ReadLine());

        bmi = w * 703 / Math.Pow(h, 2);

        if (bmi < 15) {
            Console.WriteLine("Very severely underweight");
        }
        else if (bmi < 16) {
            Console.WriteLine("Severely underweight");
        }
        else if (bmi < 18.5) {
            Console.WriteLine("Underweight");
        }
        else if (bmi < 25) {
            Console.WriteLine("Normal");
        }
    }
}
```

```

    else if (bmi < 30) {
        Console.WriteLine("Overweight");
    }
    else if (bmi < 35) {
        Console.WriteLine("Severely overweight");
    }
    else {
        Console.WriteLine("Very severely overweight");
    }
}
}

```

22. Solution

```

const double TAX_RATE = 0.10;

static void Main(string[] args) {
    int water;
    double total;

    Console.Write("Enter water consumption (in cubic feet): ");
    water = Int32.Parse(Console.ReadLine());

    if (water < 0) {
        Console.WriteLine("Error! You entered a negative value");
    }
    else {
        if (water <= 10) {
            total = water * 3;
        }
        else if (water <= 20) {
            total = 10 * 3 + (water - 10) * 5;
        }
        else if (water <= 35) {
            total = 10 * 3 + 10 * 5 + (water - 20) * 7;
        }
        else {
            total = 10 * 3 + 10 * 5 + 15 * 7 + (water - 35) * 9;
        }

        total = total + total * TAX_RATE;
        Console.WriteLine("Total amount to pay (taxes included): " + total);
    }
}

```

23. Solution

```

static void Main(string[] args) {
    int children;
    double income, tax;

    Console.Write("Enter taxable income: ");
    income = Double.Parse(Console.ReadLine());
}

```

```
Console.Write("Enter number of children: ");
children = Int32.Parse(Console.ReadLine());

if (income <= 8000) {
    tax = income * 0.10;
}
else if (income <= 30000) {
    tax = 8000 * 0.10 + (income - 8000) * 0.15;
}
else if (income <= 70000) {
    tax = 8000 * 0.10 + 22000 * 0.15 + (income - 30000) * 0.25;
}
else {
    tax = 8000 * 0.10 + 22000 * 0.15 + 40000 * 0.25 + (income - 70000) * 0.30;
}

if (children > 0) {
    tax = tax - tax * 0.02;
}
Console.WriteLine("Tax: " + tax);
}
```

24. Solution

```
static void Main(string[] args) {
    double wind;

    Console.Write("Enter wind speed (in miles/hour): ");
    wind = Double.Parse(Console.ReadLine());

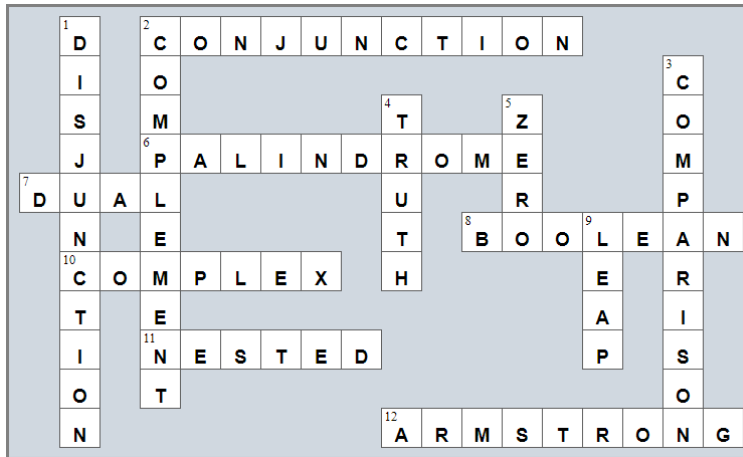
    if (wind < 0) {
        Console.WriteLine("Error! You entered a negative value");
    }
    else {
        if (wind < 1) {
            Console.WriteLine("Beaufort: 0\nCalm");
        }
        else if (wind < 4) {
            Console.WriteLine("Beaufort: 1\nLight air");
        }
        else if (wind < 8) {
            Console.WriteLine("Beaufort: 2\nLight breeze");
        }
        else if (wind < 13) {
            Console.WriteLine("Beaufort: 3\nGentle breeze");
        }
        else if (wind < 18) {
            Console.WriteLine("Beaufort: 4\nModerate breeze");
        }
        else if (wind < 25) {
            Console.WriteLine("Beaufort: 5\nFresh breeze");
        }
    }
}
```

```
    else if (wind < 31) {  
        Console.WriteLine("Beaufort: 6\nStrong breeze");  
    }  
    else if (wind < 39) {  
        Console.WriteLine("Beaufort: 7\nModerate gale");  
    }  
    else if (wind < 47) {  
        Console.WriteLine("Beaufort: 8\nGale");  
    }  
    else if (wind < 55) {  
        Console.WriteLine("Beaufort: 9\nStrong gale");  
    }  
    else if (wind < 64) {  
        Console.WriteLine("Beaufort: 10\nStorm");  
    }  
    else if (wind < 74) {  
        Console.WriteLine("Beaufort: 11\nViolent storm");  
    }  
    else {  
        Console.WriteLine("Beaufort: 12\nHurricane force");  
    }  
  
    if (wind < 13) {  
        Console.WriteLine("It's Fishing Day!!!");  
    }  
}
```

Review in “Decision Control Structures”

Review Crossword Puzzle

1.



Chapter 24

24.3 Review Questions: True/False

- | | |
|----------|----------|
| 1. true | 4. false |
| 2. true | 5. true |
| 3. false | |

Chapter 25

25.4 Review Questions: True/False

- | | |
|----------|-----------|
| 1. true | 9. false |
| 2. false | 10. false |
| 3. false | 11. false |
| 4. false | 12. true |
| 5. false | 13. false |
| 6. false | 14. false |
| 7. true | 15. true |
| 8. true | 16. false |

25.5 Review Questions: Multiple Choice

- | | |
|------|-------|
| 1. c | 7. c |
| 2. c | 8. b |
| 3. a | 9. b |
| 4. b | 10. d |
| 5. d | 11. a |
| 6. b | 12. d |

25.6 Review Exercises

1. Solution

```
static void Main(string[] args) {
    int i;

    i = 3;
    do {
        Console.WriteLine(i);
        i--;
    } while (i > 0);
    Console.WriteLine("The end");
}
```

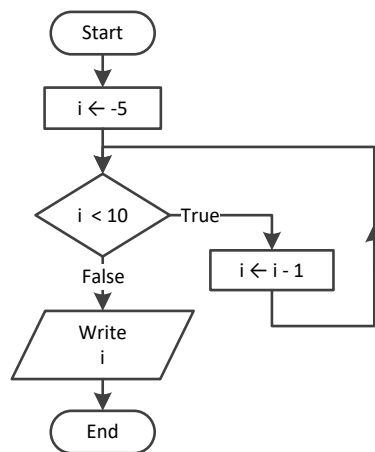
2. Solution

Step	Statement	i	x
1	i = 3	3	?
2	x = 0	3	0
3	while (i >= 0)	true	
4	i--	2	0
5	x += i	2	2
6	while (i >= 0)	true	
7	i--	1	2
8	x += i	1	3

9	while (i >= 0)	true	
10	i--	0	3
11	x += i	0	3
12	while (i >= 0)	true	
13	i--	-1	3
14	x += i	-1	2
15	while (i >= 0)	false	
16	Console.WriteLine(x)	It displays: 2	

It performs 4 iterations

3. Solution



Step	Statement	Notes	i
1	i = -5		-5
2	while (i < 10)	true	
3	i--		-6
4	while (i < 10)	true	
5	i--		-7
6	while (i < 10)	true	
7	i--		-8
8
9

It performs an infinite number of iterations

4. Solution

Step	Statement	a	b	c	d
1	a = 2	2	?	?	?
2	while (a <= 10)	true			

3	b = a + 1	2	3	?	?
4	c = b * 2	2	3	6	?
5	d = c - b + 1	2	3	6	4
6	d == 4	true			
7	Console.WriteLine(b + ", " + c)	It displays: 3, 6			
8	a += 4	6	3	6	4
9	while (a <= 10)	true			
10	b = a + 1	6	7	6	4
11	c = b * 2	6	7	14	4
12	d = c - b + 1	6	7	14	8
13	d == 4	false			
14	d == 5	false			
15	d == 8	true			
16	Console.WriteLine(a + ", " + b)	It displays: 6, 7			
17	a += 4	10	7	14	8
18	while (a <= 10)	true			
19	b = a + 1	10	11	14	8
20	c = b * 2	10	11	22	8
21	d = c - b + 1	10	11	22	12
22	d == 4	false			
23	d == 5	false			
24	d == 8	false			
25	Console.WriteLine(a + ", " + b + ", " + d)	It displays: 10, 11, 12			
26	a += 4	14	11	22	12
27	while (a <= 10)	false			

5. Solution

Step	Statement	a	b	c	d	x
1	a = 1	1	?	?	?	?
2	b = 1	1	1	?	?	?
3	c = 0	1	1	0	?	?
4	d = 0	1	1	0	0	?
5	while (b < 2)	true				
6	x = a + b	1	1	0	0	2
7	if (x % 2 != 0)	false				
8	d = d + 1	1	1	0	1	2
9	a = b	1	1	0	1	2

10	b = c	1	0	0	1	2
11	c = d	1	0	1	1	2
12	while (b < 2)	true				
13	x = a + b	1	0	1	1	1
14	if (x % 2 != 0)	true				
15	c = c + 1	1	0	2	1	1
16	a = b	0	0	2	1	1
17	b = c	0	2	2	1	1
18	c = d	0	2	1	1	1
19	while (b < 2)	false				

6. Solution

- i. -1
- ii. 9
- iii. 0.25
- iv. -7
- v. Any value between 17 and 32
- vi. 1.4

7. Solution

Step	Statement	x	y
1	y = 5	?	5
2	x = 38	38	5
3	y *= 2	38	10
4	x++	39	10
5	Console.WriteLine(y)	It displays: 10	
6	while (y < x)	true	
7	y *= 2	39	20
8	x++	40	20
9	Console.WriteLine(y)	It displays: 20	
10	while (y < x)	true	
11	y *= 2	40	40
12	x++	41	40
13	Console.WriteLine(y)	It displays: 40	
14	while (y < x)	true	
15	y *= 2	41	80
16	x++	42	80
17	Console.WriteLine(y)	It displays: 80	

18	<code>while (y < x)</code>	false
-----------	-------------------------------	-------

8. Solution

Step	Statement	Notes	x
1	<code>x = 1</code>		1
2	<code>if (x % 2 == 0)</code>	false	
3	<code>x += 3</code>		4
4	<code>Console.WriteLine(x)</code>	It displays: 4	
5	<code>while (x < 12)</code>	true	
6	<code>if (x % 2 == 0)</code>	true	
7	<code>x++</code>		5
8	<code>Console.WriteLine(x)</code>	It displays: 5	
9	<code>while (x < 12)</code>	true	
10	<code>if (x % 2 == 0)</code>	false	
11	<code>x += 3</code>		8
12	<code>Console.WriteLine(x)</code>	It displays: 8	
13	<code>while (x < 12)</code>	true	
14	<code>if (x % 2 == 0)</code>	true	
15	<code>x++</code>		9
16	<code>Console.WriteLine(x)</code>	It displays: 9	
17	<code>while (x < 12)</code>	true	
18	<code>if (x % 2 == 0)</code>	false	
19	<code>x += 3</code>		12
20	<code>Console.WriteLine(x)</code>	It displays: 12	
21	<code>while (x < 12)</code>	false	

9. Solution

Step	Statement	x	y
1	<code>y = 2</code>	?	2
2	<code>x = 0</code>	0	2
3	<code>y = Math.Pow (y, 2)</code>	0	4
4	<code>if (x < 256)</code>	true	
5	<code>x = x + y</code>	4	
6	<code>Console.WriteLine(x + ", " + y)</code>	It displays: 4, 4	
7	<code>while (y < 65535)</code>	true	
8	<code>y = Math.Pow (y, 2)</code>	4	16
9	<code>if (x < 256)</code>	true	

10	<code>x = x + y</code>	20	16
11	<code>Console.WriteLine(x + ", " + y)</code>	It displays: 20, 16	
12	<code>while (y < 65535)</code>	true	
13	<code>y = Math.Pow (y, 2)</code>	20	256
14	<code>if (x < 256)</code>	true	
15	<code>x = x + y</code>	276	256
16	<code>Console.WriteLine(x + ", " + y)</code>	It displays: 276, 256	
17	<code>while (y < 65535)</code>	true	
18	<code>y = Math.Pow (y, 2)</code>	276	65536
19	<code>if (x < 256)</code>	false	
20	<code>Console.WriteLine(x + ", " + y)</code>	It displays: 276, 65536	
21	<code>while (y < 65535)</code>	false	

10. Solution

Step	Statement	a	b	c	d	x
1	<code>a = 2</code>	2	?	?	?	?
2	<code>b = 4</code>	2	4	?	?	?
3	<code>c = 0</code>	2	4	0	?	?
4	<code>d = 0</code>	2	4	0	0	?
5	<code>x = a + b</code>	2	4	0	0	6
6	<code>if (x % 2 != 0)</code>	false				
7	<code>else if (d % 2 == 0)</code>	true				
8	<code>d = d + 5</code>	2	4	0	5	6
9	<code>a = b</code>	4	4	0	5	6
10	<code>b = d</code>	4	5	0	5	6
11	<code>while (c < 11)</code>	true				
12	<code>x = a + b</code>	4	5	0	5	9
13	<code>if (x % 2 != 0)</code>	true				
14	<code>c = c + 5</code>	4	5	5	5	9
15	<code>a = b</code>	5	5	5	5	9
16	<code>b = d</code>	5	5	5	5	9
17	<code>while (c < 11)</code>	true				
18	<code>x = a + b</code>	5	5	5	5	10

19	if (x % 2 != 0)	false				
20	else if (d % 2 == 0)	false				
21	c = c + 3	5	5	8	5	10
22	a = b	5	5	8	5	10
23	b = d	5	5	8	5	10
24	while (c < 11)	true				
25	x = a + b	5	5	8	5	10
26	if (x % 2 != 0)	false				
27	else if (d % 2 == 0)	false				
28	c = c + 3	5	5	11	5	10
29	a = b	5	5	11	5	10
30	b = d	5	5	11	5	10
31	while (c < 11)	false				

11. Solution

- i. -1
- ii. 18
- iii. 0.5
- iv. -20
- v. 128
- vi. 11.25

12. Solution

- i. 4
- ii. -2
- iii. 2
- iv. 10

13. Solution

```
static void Main(string[] args) {
    double a, total;
    int i, n;

    n = Int32.Parse(Console.ReadLine());
    total = 0;

    i = 1;
    while (i <= n) {
        a = Double.Parse(Console.ReadLine());
        total = total + a;
    }
}
```

```
        i++;
    }

    Console.WriteLine(total);
    if (n > 0) {
        Console.WriteLine(total / n);
    }
}
```

14. Solution

```
static void Main(string[] args) {
    int a, i, n, p;
    int count = 0;

    n = Int32.Parse(Console.ReadLine());
    p = 1;

    i = 1;
    while (i <= n) {
        a = Int32.Parse(Console.ReadLine());
        if (a % 2 == 0) {
            p = p * a;
            count++;
        }
        i++;
    }

    if (count > 0) {
        Console.WriteLine(p);
    }
    else {
        Console.WriteLine("You entered no even integers");
    }
}
```

15. Solution

```
static void Main(string[] args) {
    int a, i, total;

    total = 0;

    i = 1;
    while (i <= 100) {
        a = Int32.Parse(Console.ReadLine());
        if (a % 10 == 0) {
            total = total + a;
        }
        i++;
    }
    Console.WriteLine(total);
}
```

16. Solution

```
static void Main(string[] args) {
    int a, i, total;

    total = 0;

    i = 1;
    while (i <= 20) {
        a = Int32.Parse(Console.ReadLine());
        if (a >= 100 && a <= 999) {
            total = total + a;
        }
        i++;
    }
    Console.WriteLine(total);
}
```

17. Solution

```
static void Main(string[] args) {
    double a, p;

    p = 1;

    a = Double.Parse(Console.ReadLine());
    while (a != 0) {
        p = p * a;
        a = Double.Parse(Console.ReadLine());
    }
    Console.WriteLine(p);
}
```

Step	Statement	a	p
1	p = 1	?	1.0
2	a = Double.Parse(Console.ReadLine())	3.0	1.0
3	while (a != 0)	true	
4	p = p * a	3.0	3.0
5	a = Double.Parse(Console.ReadLine())	2.0	3.0
6	while (a != 0)	true	
7	p = p * a	2.0	6.0
8	a = Double.Parse(Console.ReadLine())	9.0	6.0
9	while (a != 0)	true	
10	p = p * a	9.0	54.0
11	a = Double.Parse(Console.ReadLine())	0.0	54.0
12	while (a != 0)	false	
13	Console.WriteLine(p)	It displays: 54	

18. Solution

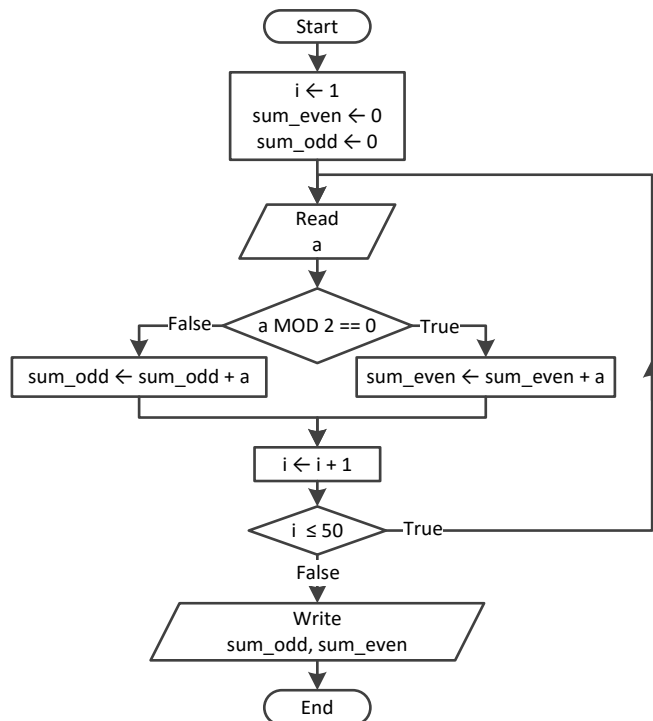
```

static void Main(string[] args) {
    int years;
    double population;

    population = 30000;

    years = 0;
    while (population <= 100000) {
        population += population * 0.03;
        years++;
    }
    Console.WriteLine(years);
}

```

19. Solution

```

static void Main(string[] args) {
    int a, i, sum_even, sum_odd;

    i = 1;
    sum_even = 0;
    sum_odd = 0;
    do {
        a = Int32.Parse(Console.ReadLine());
        if (a % 2 == 0) {
            sum_even += a;
        }
        else {
            sum_odd += a;
        }
    }
}

```



```
    }  
    i++;  
} while (i <= 50);  
Console.WriteLine(sum_even + " " + sum_odd);  
}
```

20. Solution

```
static void Main(string[] args) {  
    int a, i, n, p;  
  
    n = Int32.Parse(Console.ReadLine());  
    i = 1;  
    p = 1;  
    do {  
        a = Int32.Parse(Console.ReadLine());  
        if (a < 0) {  
            p *= a;  
        }  
        i++;  
    } while (i <= n);  
    Console.WriteLine(Math.Abs(p));  
}
```

21. Solution

```
static void Main(string[] args) {  
    int a, i, p;  
  
    i = 1;  
    p = 1;  
    do {  
        Console.Write("Enter an integer: ");  
        a = Int32.Parse(Console.ReadLine());  
        if (a >= 500 && a <= 599) {  
            p *= a;  
        }  
        i++;  
    } while (i <= 5);  
    Console.WriteLine(p);  
}
```

22. Solution

```
static void Main(string[] args) {  
    double population;  
    int years;  
  
    population = 50000;  
  
    years = 0;  
    do {  
        population -= population * 0.10;  
    }  
}
```

```
        years++;  
    } while (population >= 20000);  
    Console.WriteLine(years);  
}
```

Chapter 26

26.3 Review Questions: True/False

- | | |
|----------|-----------|
| 1. true | 7. false |
| 2. true | 8. true |
| 3. false | 9. false |
| 4. false | 10. false |
| 5. false | 11. false |
| 6. true | 12. false |

26.4 Review Questions: Multiple Choice

- | | |
|------|-------|
| 1. c | 8. b |
| 2. d | 9. c |
| 3. d | 10. b |
| 4. b | 11. d |
| 5. a | 12. d |
| 6. b | 13. d |
| 7. a | 14. b |

26.5 Review Exercises

1. Solution

Step	Statement	a	b	j
1	a = 0	0	?	?
2	b = 0	0	0	?
3	j = 0	0	0	0
4	j <= 8	true		
5	if (j < 5)	true		
6	b++	0	1	0
7	j += 2	0	1	2
8	j <= 8	true		
9	if (j < 5)	true		
10	b++	0	2	2
11	j += 2	0	2	4
12	j <= 8	true		
13	if (j < 5)	true		
14	b++	0	3	4
15	j += 2	0	3	6
16	j <= 8	true		
17	if (j < 5)	false		
18	a += j - 1	5	3	6

19	j += 2	5	3	8
20	j <= 8	true		
21	if (j < 5)	false		
22	a += j - 1	12	3	8
23	j += 2	12	3	10
24	j <= 8	false		
25	Console.WriteLine(a + ", " + b)	It displays: 12, 3		

2. Solution

For input value of 10

Step	Statement	a	b	j
1	a = Int32.Parse(Console.ReadLine())	10	?	?
2	b = a	10	10	?
3	j = a - 5	10	10	5
4	j <= a	true		
5	if (j % 2 != 0)	true		
6	b = a + j + 5	10	20	5
7	j += 2	10	20	7
8	j <= a	true		
9	if (j % 2 != 0)	true		
10	b = a + j + 5	10	22	7
11	j += 2	10	22	9
12	j <= a	true		
13	if (j % 2 != 0)	true		
14	b = a + j + 5	10	24	9
15	j += 2	10	24	11
16	j <= a	false		
17	Console.WriteLine(b)	It displays: 24		

For input value of 21

Step	Statement	a	b	j
1	a = Int32.Parse(Console.ReadLine())	21	?	?
2	b = a	21	21	?
3	j = a - 5	21	21	16
4	j <= a	true		
5	if (j % 2 != 0)	false		
6	b = a + j + 5	21	5	16
7	j += 2	21	5	18

8	j <= a	true		
9	if (j % 2 != 0)	false		
10	b = a + j + 5	21	3	18
11	j += 2	21	3	20
12	j <= a	true		
13	if (j % 2 != 0)	false		
14	b = a + j + 5	21	1	20
15	j += 2	21	1	22
16	j <= a	false		
17	Console.WriteLine(b)	It displays: 1		

3. Solution

For input value of 12

Step	Statement	a	x	y	j
1	a = Int32.Parse(Console.ReadLine())	12	?	?	?
2	j = 2	12	?	?	2
3	j <= a - 1	true			
4	x = j * 3 + 3	12	9	?	2
5	y = j * 2 + 10	12	9	14	2
6	if (y - x > 0 x > 30)	true			
7	y *= 2	12	9	28	2
8	x += 4	12	13	28	2
9	Console.WriteLine(x + ", " + y)	It displays: 13, 28			
10	j += 3	12	13	28	5
11	j <= a - 1	true			
12	x = j * 3 + 3	12	18	28	5
13	y = j * 2 + 10	12	18	20	5
14	if (y - x > 0 x > 30)	true			
15	y *= 2	12	18	40	5
16	x += 4	12	22	40	5
17	Console.WriteLine(x + ", " + y)	It displays: 22, 40			
18	j += 3	12	22	40	8
19	j <= a - 1	true			
20	x = j * 3 + 3	12	27	40	8
21	y = j * 2 + 10	12	27	26	8
22	if (y - x > 0 x > 30)	false			
23	x += 4	12	31	26	8

24	Console.WriteLine(x + ", " + y)	It displays: 31, 26			
25	j += 3	12	31	26	11
26	j <= a - 1	true			
27	x = j * 3 + 3	12	36	26	11
28	y = j * 2 + 10	12	36	32	11
29	if (y - x > 0 x > 30)	true			
30	y *= 2	12	36	64	11
31	x += 4	12	40	64	11
32	Console.WriteLine(x + ", " + y)	It displays: 40, 64			
33	j += 3	12	40	64	14
34	j <= a - 1	false			

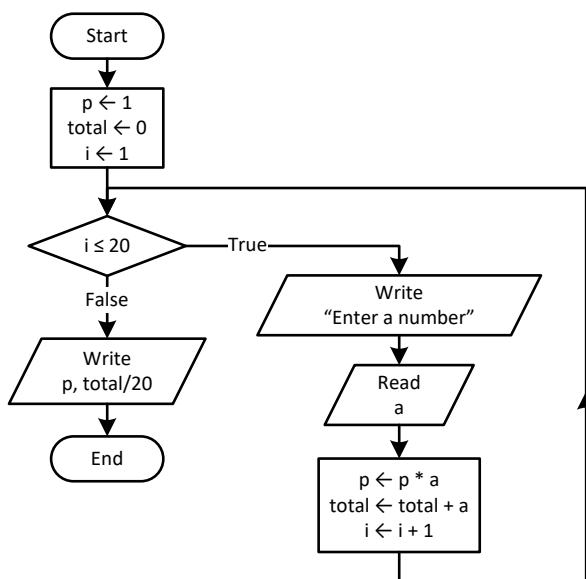
4. Solution

- i. 9
- ii. Any value greater than or equal to 2 and less than 2.5 ($2 \leq x < 2.5$)
- iii. -7 (or -6)
- iv. -1

5. Solution

It displays: sueZ

6. Solution



```

static void Main(string[] args) {
    double a, p, total;
    int i;

    p = 1;

```

```
total = 0;
for (i = 1 ; i <= 20; i++) {
    Console.Write("Enter a number: ");
    a = Double.Parse(Console.ReadLine());
    p = p * a;
    total = total + a;
}
Console.WriteLine(p);
Console.WriteLine(total / 20);
}
```

7. Solution

```
static void Main(string[] args) {
    double i;

    for (i = 0 ; i <= 360; i += 0.5) {
        Console.WriteLine(Math.Sin(i * Math.PI / 180));
    }
}
```

8. Solution

```
static void Main(string[] args) {
    int deg, i;

    Console.Write("Enter degrees: ");
    deg = Int32.Parse(Console.ReadLine());
    for (i = 0 ; i <= deg; i++) {
        Console.WriteLine(Math.Cos(i * Math.PI / 180));
    }
}
```

9. Solution

```
static void Main(string[] args) {
    int i, s;

    s = 0;
    for (i = 1; i <= 99; i += 2) {
        s += i;
    }
    Console.WriteLine(s);
}
```

10. Solution

```
static void Main(string[] args) {
    int i, n;
    double p;

    n = Int32.Parse(Console.ReadLine());
    p = 1;
```

```

    for (i = 2; i <= 2 * n; i += 2) {
        p *= Math.Pow(i, i - 1);
    }
    Console.WriteLine(p);
}

```

11. Solution

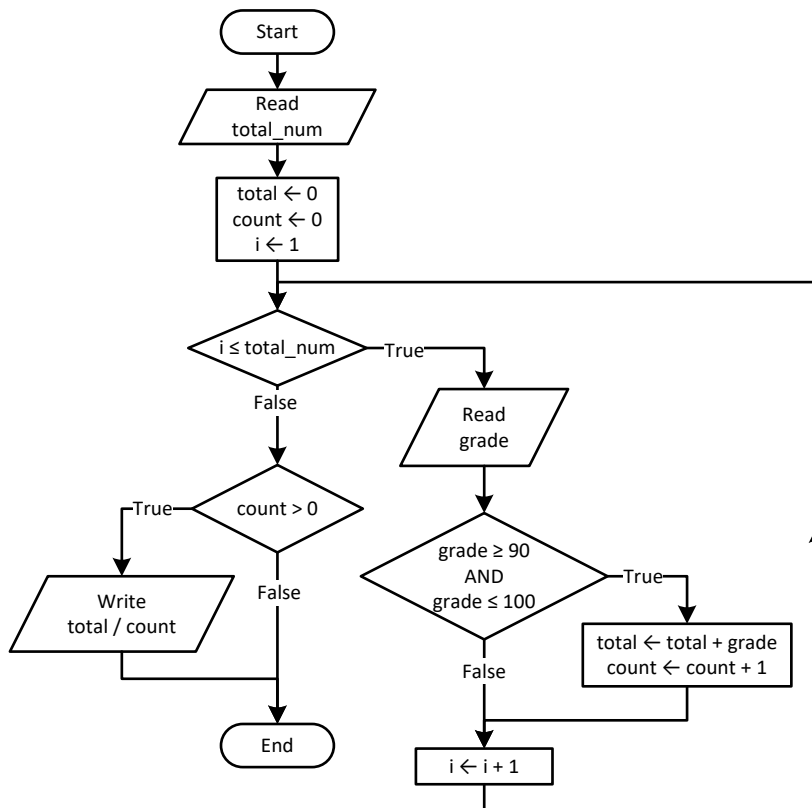
```

static void Main(string[] args) {
    int i, offset, s;

    s = 0;
    i = 1;
    offset = 0;
    while (i <= 191) {
        s += i;
        offset++;
        i += offset;
    }
    Console.WriteLine(s);
}

```

12. Solution



```

static void Main(string[] args) {
    int count, grade, i, total_num, total;

    total_num = Int32.Parse(Console.ReadLine());
}

```

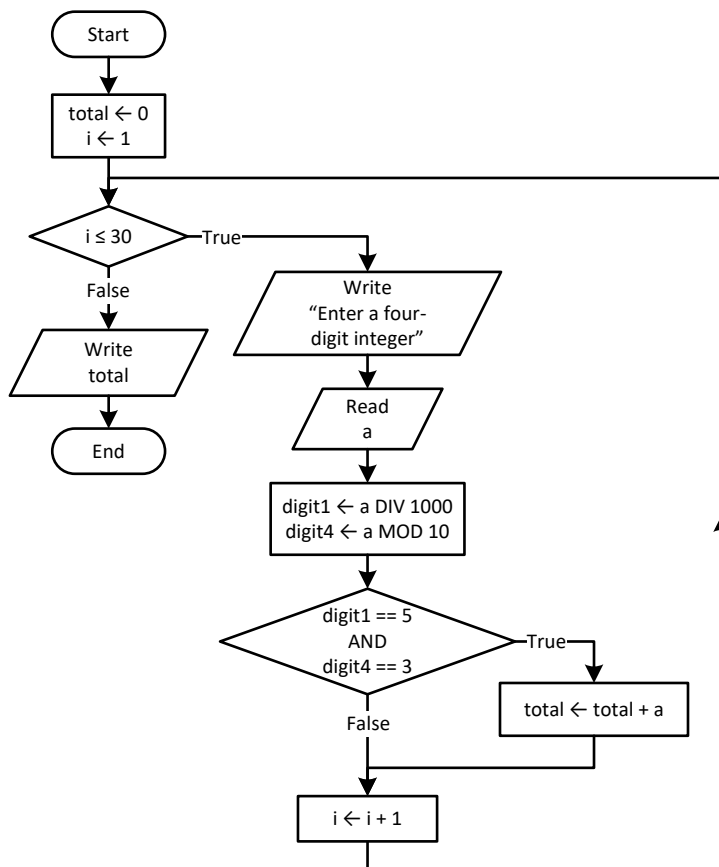


```

total = 0;
count = 0;
for (i = 1; i <= total_num; i++) {
    grade = Int32.Parse(Console.ReadLine());
    if (grade >= 90 && grade <= 100) {
        total += grade;
        count++;
    }
}
if (count > 0) {
    Console.WriteLine(total / (double)count);
}
}

```

13. Solution



```

static void Main(string[] args) {
    int a, digit1, digit4, i, total;

    total = 0;
    for (i = 1; i <= 30; i++) {
        Console.Write("Enter a four-digit integer: ");
        a = Int32.Parse(Console.ReadLine());
        digit1 = (int)(a / 1000);
        digit4 = a % 10;
        if (digit1 == 5 && digit4 == 3) {

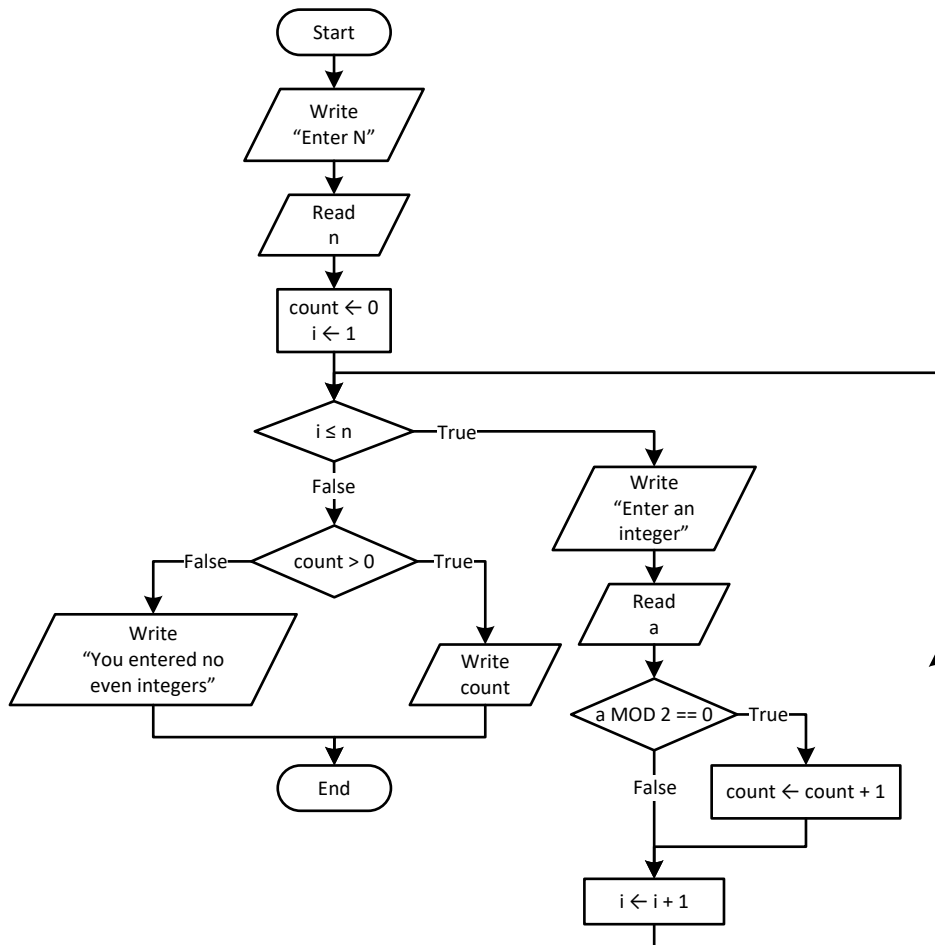
```

```

        total += a;
    }
}
Console.WriteLine(total);
}

```

14. Solution



```

static void Main(string[] args) {
    int a, count, i, n;

    Console.Write("Enter N: ");
    n = Int32.Parse(Console.ReadLine());
    count = 0;
    for (i = 1; i <= n; i++) {
        Console.Write("Enter an integer: ");
        a = Int32.Parse(Console.ReadLine());
        if (a % 2 == 0) {
            count++;
        }
    }
    if (count > 0) {
        Console.WriteLine(count);
    }
}

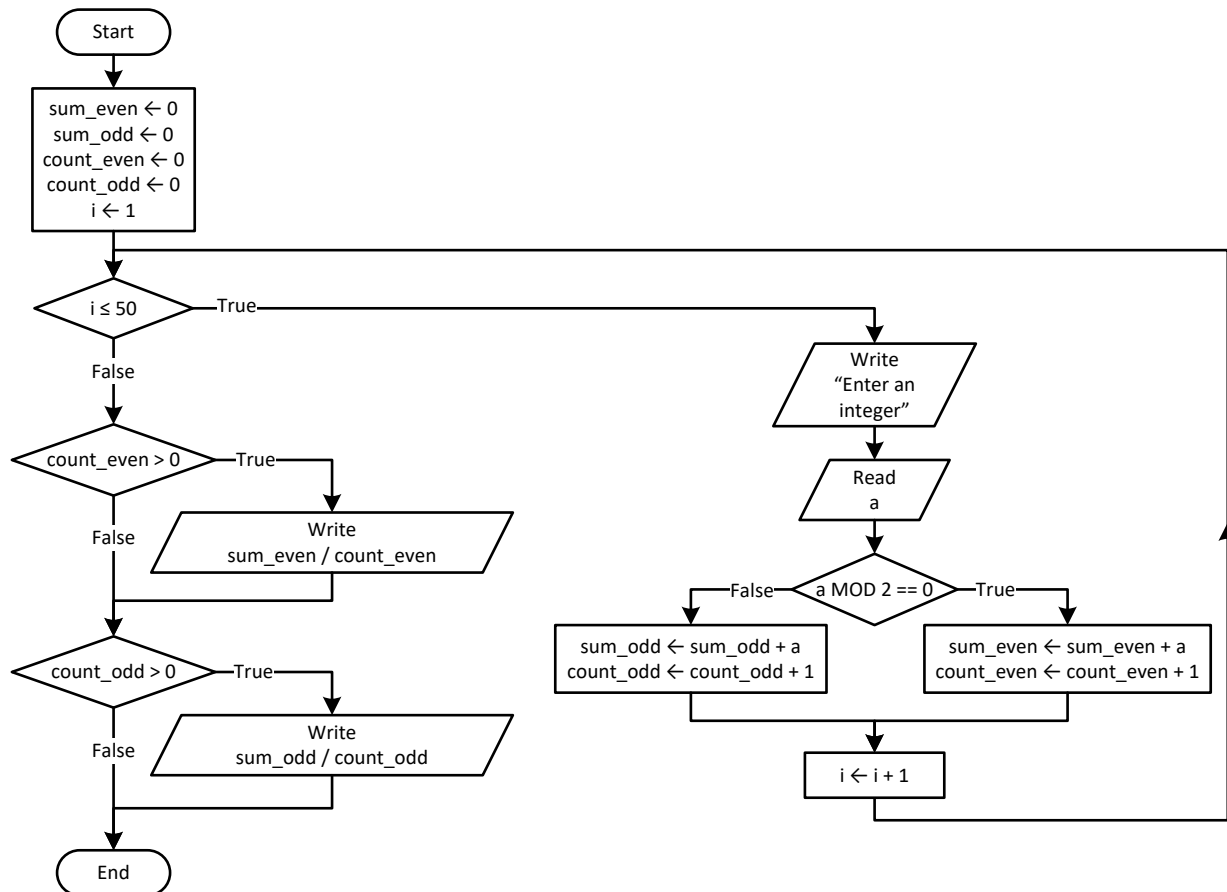
```

```

    }
    else {
        Console.WriteLine("You entered no even integers");
    }
}

```

15. Solution



```

static void Main(string[] args) {
    int a, count_even, count_odd, i, sum_even, sum_odd;

    sum_even = 0;
    sum_odd = 0;
    count_even = 0;
    count_odd = 0;
    for (i = 1; i <= 50; i++) {
        Console.Write("Enter an integer: ");
        a = Int32.Parse(Console.ReadLine());
        if (a % 2 == 0) {
            sum_even += a;
            count_even++;
        }
        else {
            sum_odd += a;
            count_odd++;
        }
    }
}

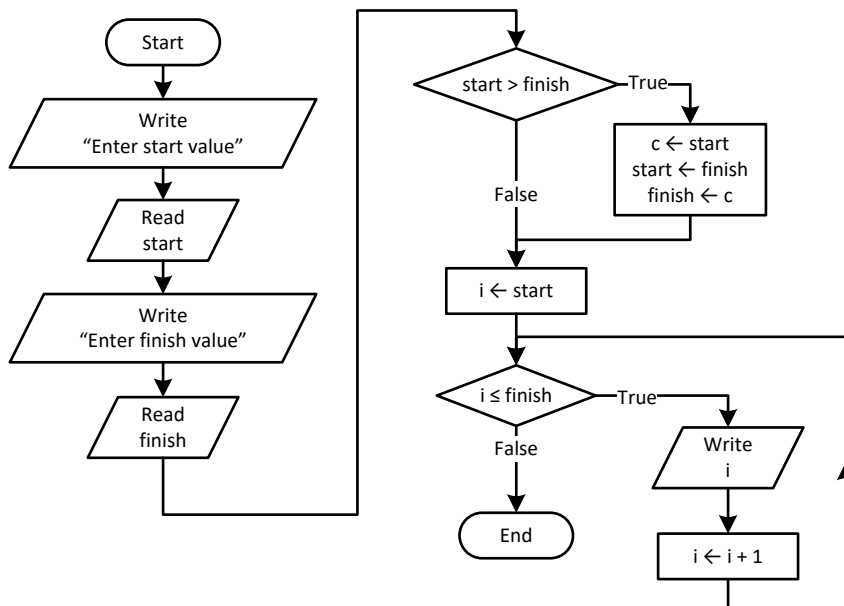
```

```

    }
}
if (count_even > 0) {
    Console.WriteLine(sum_even / (double)count_even);
}
if (count_odd > 0) {
    Console.WriteLine(sum_odd / (double)count_odd);
}
}

```

16. Solution



```

static void Main(string[] args) {
    int c, finish, i, start;

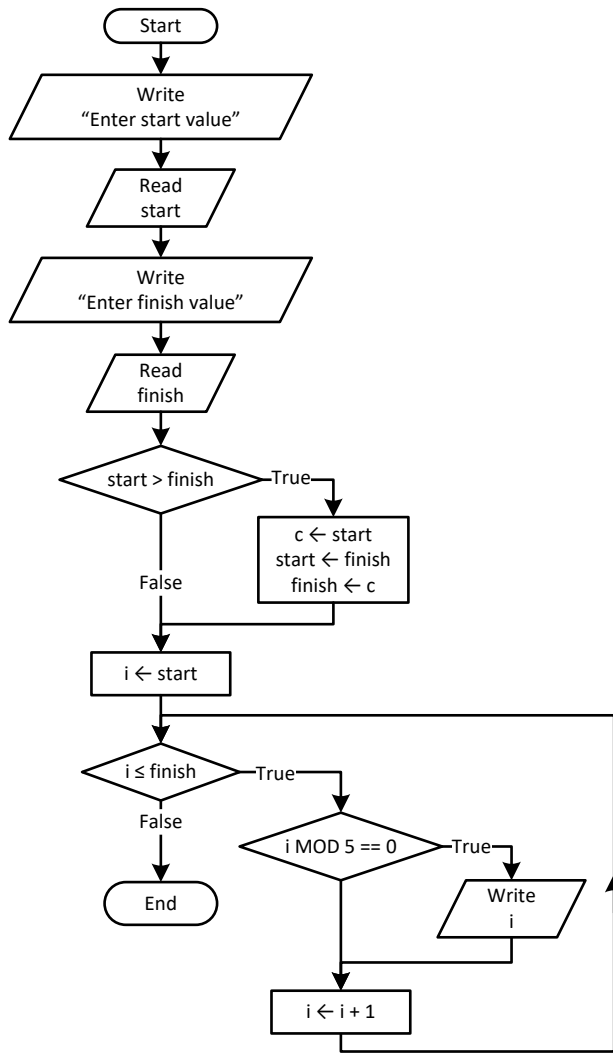
    Console.Write("Enter start value: ");
    start = Int32.Parse(Console.ReadLine());
    Console.Write("Enter finish value: ");
    finish = Int32.Parse(Console.ReadLine());

    if (start > finish) {
        c = start;
        start = finish;
        finish = c;
    }

    for (i = start; i <= finish; i++) {
        Console.WriteLine(i);
    }
}

```

17. Solution



```

static void Main(string[] args) {
    int c, finish, i, start;

    Console.Write("Enter start value: ");
    start = Int32.Parse(Console.ReadLine());
    Console.Write("Enter finish value: ");
    finish = Int32.Parse(Console.ReadLine());

    if (start > finish) {
        c = start;
        start = finish;
        finish = c;
    }

    for (i = start; i <= finish; i++) {
        if (i % 5 == 0) {
            Console.WriteLine(i);
        }
    }
}

```

```

    }
}

```

18. Solution

First approach

```

static void Main(string[] args) {
    int exp, i;
    double p, b;

    Console.Write("Enter a value for base: ");
    b = Double.Parse(Console.ReadLine());
    Console.Write("Enter an integer for exponent: ");
    exp = Int32.Parse(Console.ReadLine());

    p = 1;
    if (exp >= 0) {
        for (i = 1; i <= exp; i++) {
            p *= b;
        }
    }
    else {
        for (i = 1; i <= -exp; i++) {
            p *= 1 / b;
        }
    }
    Console.WriteLine(p);
}

```

Second approach

```

static void Main(string[] args) {
    int exp, i;
    double p, b;

    Console.Write("Enter a value for base: ");
    b = Double.Parse(Console.ReadLine());
    Console.Write("Enter an integer for exponent: ");
    exp = Int32.Parse(Console.ReadLine());

    p = 1;
    for (i = 1; i <= Math.Abs(exp); i++) {
        p *= b;
    }
    if (exp < 0) {
        p = 1 / p;
    }
    Console.WriteLine(p);
}

```

19. Solution

```

static void Main(string[] args) {
    int count, i, words;

```

```
string msg, character;

Console.Write("Enter a message: ");
msg = Console.ReadLine();

count = 0;
for (i = 0; i <= msg.Length - 1; i++) {
    character = "" + msg[i];
    if (character == " ") {
        count++;
    }
}
words = count + 1;

Console.WriteLine("The message entered contains " + words + " words");
}
```

20. Solution

```
static void Main(string[] args) {
    int characters, count, i, words;
    string msg, character;

    Console.Write("Enter a message: ");
    msg = Console.ReadLine();

    characters = msg.Length;
    count = 0;
    for (i = 0; i <= characters - 1; i++) {
        character = "" + msg[i];
        if (character == " ") {
            count++;
        }
    }

    words = count + 1;
    Console.Write("The average number of letters in each word is ");
    Console.WriteLine((characters - count) / (double)words);
}
```

21. Solution

```
static void Main(string[] args) {
    string message;
    char character;
    string consonants = "BCDFGHJKLMNPQRSTVWXYZ";
    int i, count;

    Console.Write("Enter an English message: ");
    message = Console.ReadLine().ToUpper();

    count = 0;
    for (i = 0; i <= message.Length - 1; i++) {
```

```
        character = message[i];

        if (consonants.IndexOf(character) != -1) { //If character is found in consonants
            count++;
        }
    }
    Console.WriteLine("Consonants: " + count);
}
```

22. Solution

```
static void Main(string[] args) {
    string message;
    char character;
    string vowels = "AEIOU";
    string consonants = "BCDFGHJKLMNPQRSTVWZYX";
    string digits = "0123456789";
    int i, countv, countc, countd;

    Console.Write("Enter an English message: ");
    message = Console.ReadLine().ToUpper();

    countv = countc = countd = 0;
    for (i = 0; i <= message.Length - 1; i++) {
        character = message[i];

        if (vowels.IndexOf(character) != -1) { //If character is found in vowels
            countv++;
        }
        else if (consonants.IndexOf(character) != -1) { //If character is found in consonants
            countc++;
        }
        else if (digits.IndexOf(character) != -1) { //If character is found in digits
            countd++;
        }
    }
    Console.WriteLine("Vowels: " + countv);
    Console.WriteLine("Consonants: " + countc);
    Console.WriteLine("Digits: " + countd);
}
```


Chapter 27

27.3 Review Questions: True/False

- | | |
|----------|----------|
| 1. true | 6. false |
| 2. true | 7. true |
| 3. false | 8. true |
| 4. true | 9. true |
| 5. true | 10. true |

27.4 Review Questions: Multiple Choice

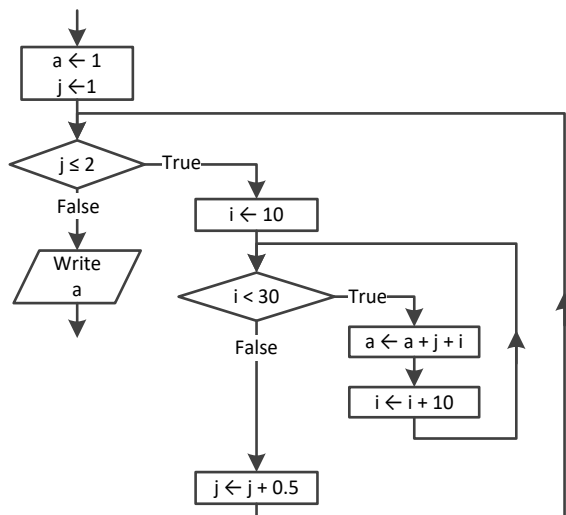
- | | |
|------|------|
| 1. b | 4. a |
| 2. a | 5. b |
| 3. c | |

27.5 Review Exercises

1. Solution

- 10
- A value greater than or equal to 4.5 and less than 5.0
- 7 (or -6)
- 138 (or 139)

2. Solution



Step	Statement	a	i	j
1	a = 1	1	?	?
2	j = 1	1	?	1
3	j ≤ 2	true		
4	i = 10	1	10	1
5	i < 30	true		

6	<code>a = a + j + i</code>	12	10	1
7	<code>i += 10</code>	12	20	1
8	<code>i < 30</code>	true		
9	<code>a = a * j + i</code>	33	20	1
10	<code>i += 10</code>	33	30	1
11	<code>i < 30</code>	false		
12	<code>j += 0.5</code>	33	30	1.5
13	<code>j <= 2</code>	true		
14	<code>i = 10</code>	33	10	1.5
15	<code>i < 30</code>	true		
16	<code>a = a + j + i</code>	44.5	10	1.5
17	<code>i += 10</code>	44.5	20	1.5
18	<code>i < 30</code>	true		
19	<code>a = a * j + i</code>	66	20	1.5
20	<code>i += 10</code>	66	30	1.5
21	<code>i < 30</code>	false		
22	<code>j += 0.5</code>	66	30	2
23	<code>j <= 2</code>	true		
24	<code>i = 10</code>	66	10	2
25	<code>i < 30</code>	true		
26	<code>a = a + j + i</code>	78	10	2
27	<code>i += 10</code>	78	20	2
28	<code>i < 30</code>	true		
29	<code>a = a * j + i</code>	100	20	2
30	<code>i += 10</code>	100	30	2
31	<code>i < 30</code>	false		
32	<code>j += 0.5</code>	100	30	2.5
33	<code>j <= 2</code>	false		
34	<code>Console.WriteLine(a)</code>	It displays: 100		

3. Solution

Step	Statement	s	i	j
1	<code>s = 0</code>	0	?	?
2	<code>i = 1</code>	0	1	?
3	<code>i <= 4</code>	true		
4	<code>j = 3</code>	0	1	3
5	<code>j >= i</code>	true		

6	<code>s = s + i * j</code>	3	1	3
7	<code>j--</code>	3	1	2
8	<code>j >= i</code>	true		
9	<code>s = s + i * j</code>	5	1	2
10	<code>j--</code>	5	1	1
11	<code>j >= i</code>	true		
12	<code>s = s + i * j</code>	6	1	1
13	<code>j--</code>	6	1	0
14	<code>j >= i</code>	false		
15	<code>i++</code>	6	2	0
16	<code>i <= 4</code>	true		
17	<code>j = 3</code>	6	2	3
18	<code>j >= i</code>	true		
19	<code>s = s + i * j</code>	12	2	3
20	<code>j--</code>	12	2	2
21	<code>j >= i</code>	true		
22	<code>s = s + i * j</code>	16	2	2
23	<code>j--</code>	16	2	1
24	<code>j >= i</code>	false		
25	<code>i++</code>	16	3	1
26	<code>i <= 4</code>	true		
27	<code>j = 3</code>	16	3	3
28	<code>j >= i</code>	true		
29	<code>s = s + i * j</code>	25	3	3
30	<code>j--</code>	25	3	2
31	<code>j >= i</code>	false		
32	<code>i++</code>	25	4	2
33	<code>i <= 4</code>	true		
34	<code>j = 3</code>	25	4	3
35	<code>j >= i</code>	false		
36	<code>i++</code>	25	5	3
37	<code>i <= 4</code>	false		
38	<code>Console.WriteLine(s)</code>	It displays: 25		

The statement `s = s + i * j` is executed 6 times

4. Solution

For input value of "NO"

Step	Statement	s	y	i	ans
1	s = 1	1	?	?	?
2	y = 25	1	25	?	?
3	i = 1	1	25	1	?
4	i <= 3	true			
5	s = s + y	26	25	1	?
6	y -= 5	26	20	1	?
7	i++	26	20	2	?
8	i <= 3	true			
9	s = s + y	46	20	2	?
10	y -= 5	46	15	2	?
11	i++	46	15	3	?
12	i <= 3	true			
13	s = s + y	61	15	3	?
14	y -= 5	61	10	3	?
15	i++	61	10	4	?
16	i <= 3	false			
17	ans = Console.ReadLine()	61	10	4	"NO"
18	while (ans == "YES")	false			
19	Console.WriteLine(s)	It displays: 61			

For input values of "YES", "NO"

Step	Statement	s	y	i	ans
1	s = 1	1	?	?	?
2	y = 25	1	25	?	?
3	i = 1	1	25	1	?
4	i <= 3	true			
5	s = s + y	26	25	1	?
6	y -= 5	26	20	1	?
7	i++	26	20	2	?
8	i <= 3	true			
9	s = s + y	46	20	2	?
10	y -= 5	46	15	2	?
11	i++	46	15	3	?
12	i <= 3	true			
13	s = s + y	61	15	3	?
14	y -= 5	61	10	3	?
15	i++	61	10	4	?

16	<code>i <= 3</code>	false			
17	<code>ans = Console.ReadLine()</code>	61	10	4	"YES"
18	<code>while (ans == "YES")</code>	true			
19	<code>i = 1</code>	61	10	1	"YES"
20	<code>i <= 3</code>	true			
21	<code>s = s + y</code>	71	10	1	"YES"
22	<code>y -= 5</code>	71	5	1	"YES"
23	<code>i++</code>	71	5	2	"YES"
24	<code>i <= 3</code>	true			
25	<code>s = s + y</code>	76	5	2	"YES"
26	<code>y -= 5</code>	76	0	2	"YES"
27	<code>i++</code>	76	0	3	"YES"
28	<code>i <= 3</code>	true			
29	<code>s = s + y</code>	76	0	3	"YES"
30	<code>y -= 5</code>	76	-5	3	"YES"
31	<code>i++</code>	76	-5	4	"YES"
32	<code>i <= 3</code>	false			
33	<code>ans = Console.ReadLine()</code>	76	-5	4	"NO"
34	<code>while (ans == "YES")</code>	false			
35	<code>Console.WriteLine(s)</code>	It displays: 76			

For input values of "YES", "YES", "NO"

Step	Statement	s	y	i	ans
1	<code>s = 1</code>	1	?	?	?
2	<code>y = 25</code>	1	25	?	?
3	<code>i = 1</code>	1	25	1	?
4	<code>i <= 3</code>	true			
5	<code>s = s + y</code>	26	25	1	?
6	<code>y -= 5</code>	26	20	1	?
7	<code>i++</code>	26	20	2	?
8	<code>i <= 3</code>	true			
9	<code>s = s + y</code>	46	20	2	?
10	<code>y -= 5</code>	46	15	2	?
11	<code>i++</code>	46	15	3	?
12	<code>i <= 3</code>	true			
13	<code>s = s + y</code>	61	15	3	?
14	<code>y -= 5</code>	61	10	3	?
15	<code>i++</code>	61	10	4	?

16	i <= 3	false			
17	ans = Console.ReadLine()	61	10	4	"YES"
18	while (ans == "YES")	true			
19	i = 1	61	10	1	"YES"
20	i <= 3	true			
21	s = s + y	71	10	1	"YES"
22	y -= 5	71	5	1	"YES"
23	i++	71	5	2	"YES"
24	i <= 3	true			
25	s = s + y	76	5	2	"YES"
26	y -= 5	76	0	2	"YES"
27	i++	76	0	3	"YES"
28	i <= 3	true			
29	s = s + y	76	0	3	"YES"
30	y -= 5	76	-5	3	"YES"
31	i++	76	-5	4	"YES"
32	i <= 3	false			
33	ans = Console.ReadLine()	76	-5	4	"YES"
34	while (ans == "YES")	true			
35	i = 1	76	-5	1	"YES"
36	i <= 3	true			
37	s = s + y	71	-5	1	"YES"
38	y -= 5	71	-10	1	"YES"
39	i++	71	-10	2	"YES"
40	i <= 3	true			
41	s = s + y	61	-10	2	"YES"
42	y -= 5	61	-15	2	"YES"
43	i++	61	-15	3	"YES"
44	i <= 3	true			
45	s = s + y	46	-15	3	"YES"
46	y -= 5	46	-20	3	"YES"
47	i++	46	-20	4	"YES"
48	i <= 3	false			
49	ans = Console.ReadLine()	46	-20	4	"NO"
50	while (ans == "YES")	false			
51	Console.WriteLine(s)	It displays: 46			

5. Solution

```
static void Main(string[] args) {
    int hour, minutes;

    for (hour = 0; hour <= 23; hour++) {
        for (minutes = 0; minutes <= 59; minutes++) {
            Console.WriteLine(hour + "\t" + minutes);
        }
    }
}
```

6. Solution

```
static void Main(string[] args) {
    int i, j;

    for (i = 5; i >= 1; i--) {
        for (j = 1; j <= i; j++) {
            Console.Write(i + " ");
        }
        Console.WriteLine();
    }
}
```

7. Solution

```
static void Main(string[] args) {
    int i, j;

    for (i = 0; i <= 5; i++) {
        for (j = 0; j <= i; j++) {
            Console.Write(j + " ");
        }
        Console.WriteLine();
    }
}
```

8. Solution

```
static void Main(string[] args) {
    int i, j;

    for (i = 1; i <= 4; i++) {
        for (j = 1; j <= 10; j++) {
            Console.Write("* ");
        }
        Console.WriteLine();
    }
}
```

9. Solution

```
static void Main(string[] args) {
    int i, j, y;

    Console.Write("Enter an integer between 3 and 20: ");
    y = Int32.Parse(Console.ReadLine());

    for (i = 1; i <= y; i++) {
        for (j = 1; j <= y; j++) {
            Console.Write("* ");
        }
        Console.WriteLine();
    }
}
```

10. Solution

```
static void Main(string[] args) {
    int i, j, y;

    Console.Write("Enter an integer between 3 and 20: ");
    y = Int32.Parse(Console.ReadLine());

    for (j = 1; j <= y; j++) {
        Console.Write("* ");
    }
    Console.WriteLine();

    for (i = 1; i <= y - 2; i++) {
        Console.Write("* ");
        for (j = 1; j <= y - 2; j++) {
            Console.Write(" ");
        }
        Console.WriteLine("* ");
    }

    for (j = 1; j <= y; j++) {
        Console.Write("* ");
    }
}
```

11. Solution

```
static void Main(string[] args) {
    int i, j;

    for (i = 1; i <= 5; i++) {
        for (j = 1; j <= i; j++) {
            Console.Write("* ");
        }
        Console.WriteLine();
    }
}
```



```
    for (i = 4; i >= 1; i--) {  
        for (j = 1; j <= i; j++) {  
            Console.Write("* ");  
        }  
        Console.WriteLine();  
    }  
}
```

Chapter 28

28.8 Review Questions: True/False

- | | |
|----------|-----------|
| 1. false | 8. false |
| 2. false | 9. true |
| 3. false | 10. true |
| 4. true | 11. false |
| 5. true | 12. false |
| 6. false | 13. false |
| 7. false | 14. true |

28.9 Review Questions: Multiple Choice

- | | |
|------|------|
| 1. c | 5. a |
| 2. d | 6. c |
| 3. b | 7. c |
| 4. a | |

28.10 Review Exercises

1. Solution

```
count_names = 0;
count_not_johns = 0;
name = "";
Console.Write("Enter a name: ");
name = Console.ReadLine();
while (name != "STOP") {
    Console.Write("Enter a name: ");
    name = Console.ReadLine();
    count_names++;
    if (name != "John") {
        count_not_johns++;
    }
    Console.Write("Enter a name: ");
    name = Console.ReadLine();
}
Console.WriteLine(count_names + " names entered");
Console.WriteLine("Names other than John entered " + count_not_johns + " times");
```

2. Solution

First approach

```
static void Main(string[] args) {
    string text, character;
    bool found;
    int i;

    Console.Write("Enter a text: ");
    text = Console.ReadLine();

    found = false;
    for (i = 0; i <= text.Length - 1; i++) {
```

```
        character = "" + text[i];
        if (character == " ") {
            found = true;
            break;
        }
    }

    if (!found) {
        Console.WriteLine("One Single Word");
    }
    else {
        Console.WriteLine("Complete Sentence");
    }
}
```

Second approach

```
static void Main(string[] args) {
    string text;

    Console.Write("Enter a text: ");
    text = Console.ReadLine();

    if (text.IndexOf(" ") == -1) {
        Console.WriteLine("One Single Word");
    }
    else {
        Console.WriteLine("Complete Sentence");
    }
}
```

3. Solution

First approach

```
static void Main(string[] args) {
    string sentence, character;
    bool found;
    int i;
    string digits = "0123456789";

    Console.Write("Enter a text: ");
    sentence = Console.ReadLine();

    found = false;
    for (i = 0; i <= sentence.Length - 1; i++) {
        character = "" + sentence[i];
        if (digits.IndexOf(character) != -1) {
            found = true;
            break;
        }
    }

    if (found) {
        Console.WriteLine("The sentence contains a number");
    }
}
```

```
}
```

Second approach

```
static void Main(string[] args) {
    string sentence;
    bool found;
    int i;
    string digit;

    Console.Write("Enter a text: ");
    sentence = Console.ReadLine();

    found = false;
    for (i = 0; i <= 9; i++) {
        digit = "" + i;
        if (sentence.IndexOf(digit) != -1) {
            found = true;
            break;
        }
    }

    if (found) {
        Console.WriteLine("The sentence contains a number");
    }
}
```

4. Solution

```
Console.WriteLine("Printing all integers from 1 to 100");
i = 1;
while (i < 101) {
    Console.WriteLine(i);
    i++;
}
```

5. Solution

```
Console.WriteLine("Printing odd integers from 1 to 99");
i = 1;
while (i < 100) {
    Console.WriteLine(i);
    i += 2;
}
```

6. Solution

```
s = 0;
for (i = 1; i <= 100; i++) {
    number = Double.Parse(Console.ReadLine());
    s = s + number;
}
average = s / 100.0;
Console.WriteLine(average);
```

7. Solution

```
int i, denom;
double s;

s = 0;

denom = 1;
for (i = 1; i <= 100; i++) {
    denom *= i;
}

for (i = 1; i <= 100; i++) {
    s += i / (double)denom;
}

Console.WriteLine(s);
```

8. Solution

```
static void Main(string[] args) {
    int i, j;

    for (i = 1; i <= 4; i++) {
        for (j = 1; j <= 4; j++) {
            Console.WriteLine(i + " x " + j + " = " + (i * j));
        }
    }
}
```

9. Solution

```
static void Main(string[] args) {
    int i, j;

    Console.Write("\t|\t");
    for (i = 1; i <= 12; i++) {
        Console.Write(i + "\t");
    }
    Console.WriteLine();

    for (i = 1; i <= 12; i++) {
        Console.Write("-----");
    }
    Console.WriteLine();

    for (i = 1; i <= 12; i++) {
        Console.Write(i + "\t|\t");
        for (j = 1; j <= 12; j++) {
            Console.Write(i * j + "\t");
        }
        Console.WriteLine();
    }
}
```

10. Solution

```
static void Main(string[] args) {
    int i, j, n;

    Console.Write("Enter an integer: ");
    n = Int32.Parse(Console.ReadLine());

    Console.Write("\t|\t");
    for (i = 1; i <= n; i++) {
        Console.Write(i + "\t");
    }
    Console.WriteLine();

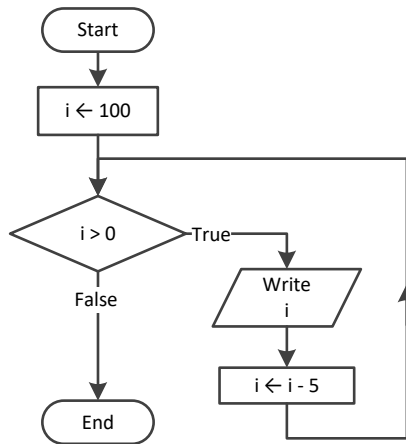
    for (i = 1; i <= n; i++) {
        Console.Write("-----");
    }
    Console.WriteLine();

    for (i = 1; i <= n; i++) {
        Console.Write(i + "\t|\t");
        for (j = 1; j <= n; j++) {
            Console.Write(i * j + "\t");
        }
        Console.WriteLine();
    }
}
```

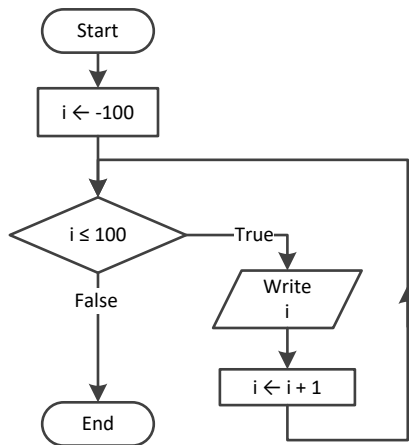
Chapter 29

29.4 Review Exercises

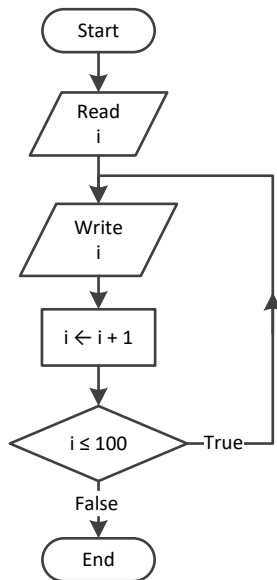
1. Solution



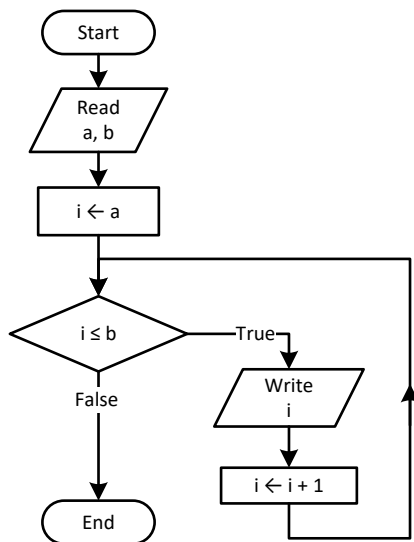
2. Solution



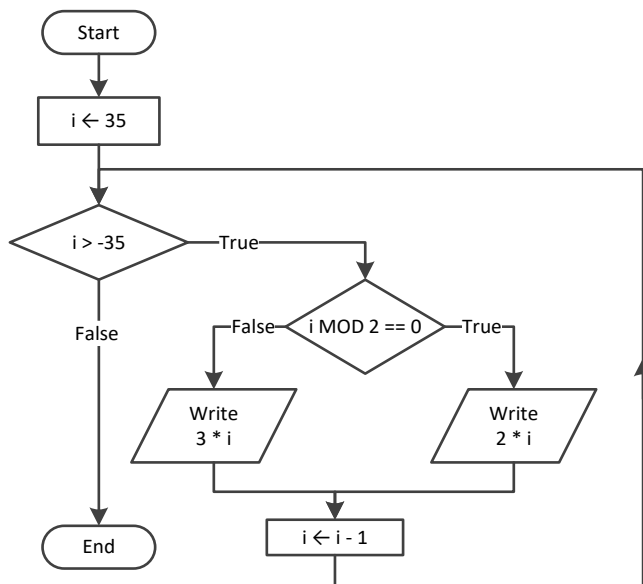
3. Solution



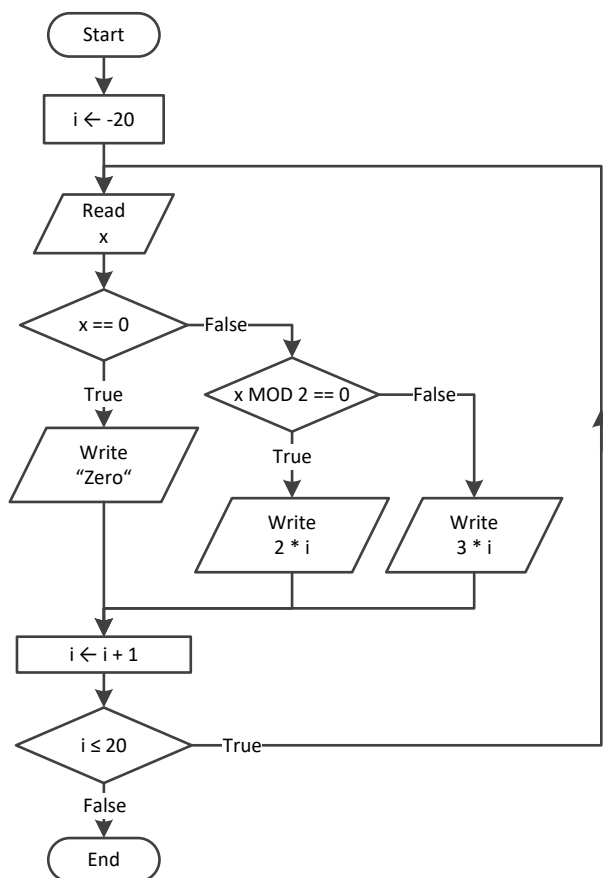
4. Solution



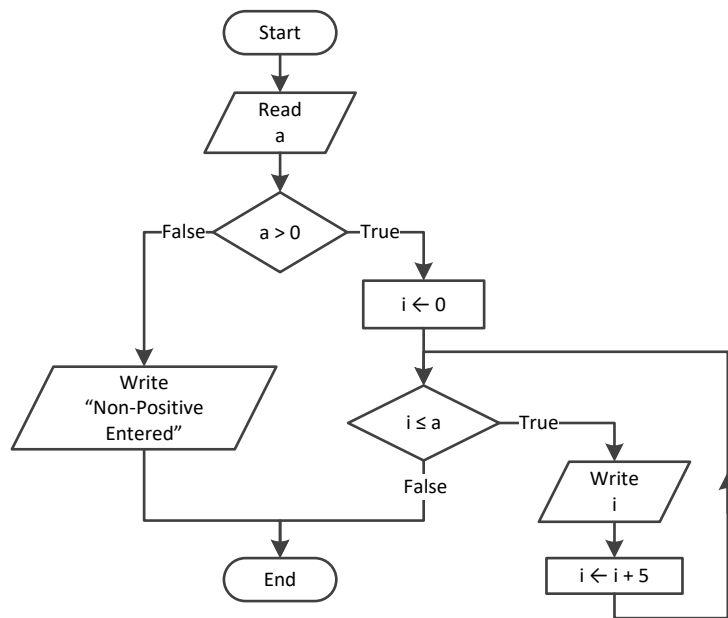
5. Solution



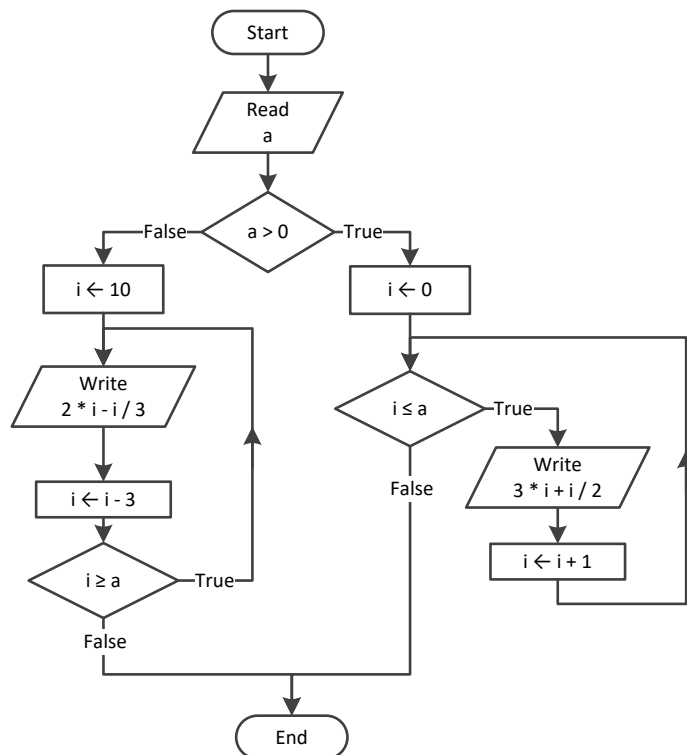
6. Solution

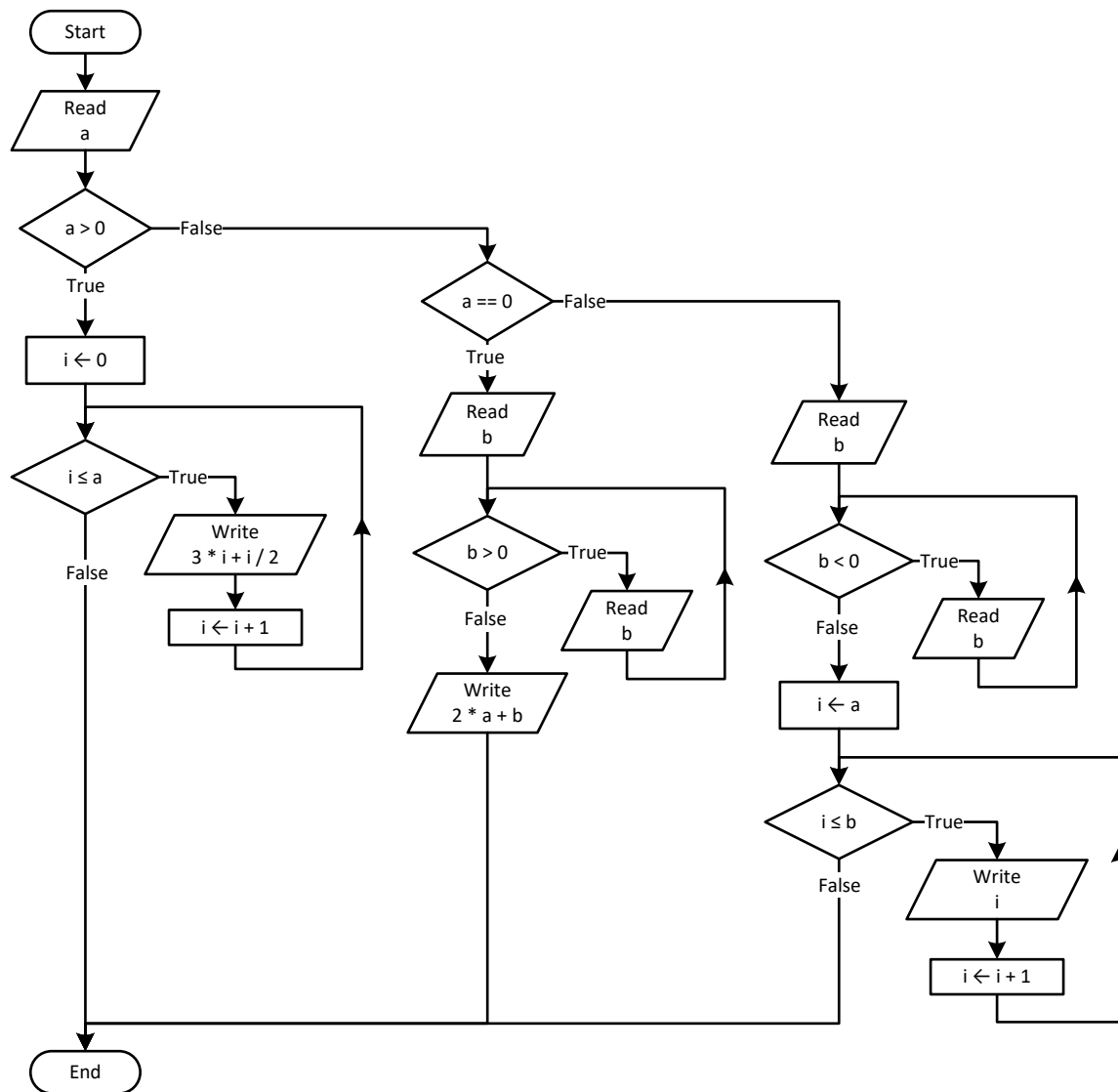


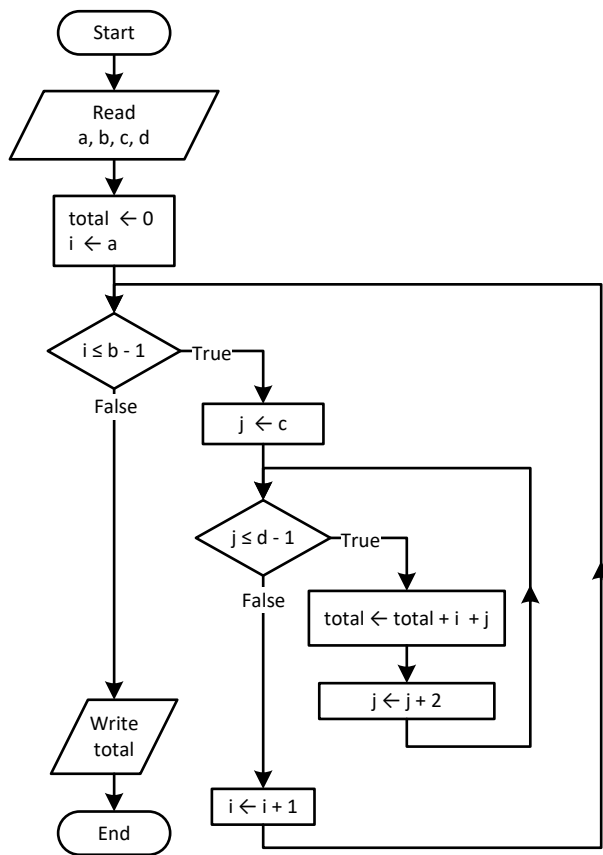
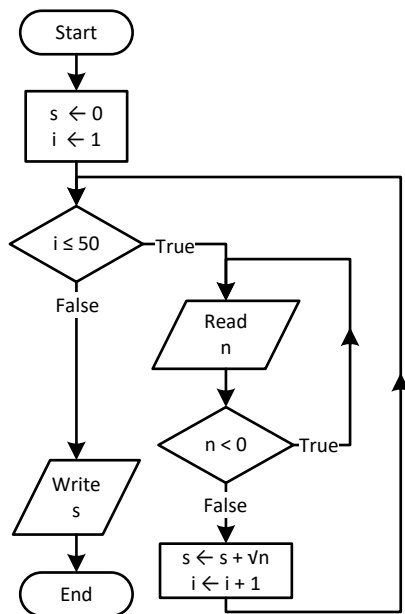
7. Solution

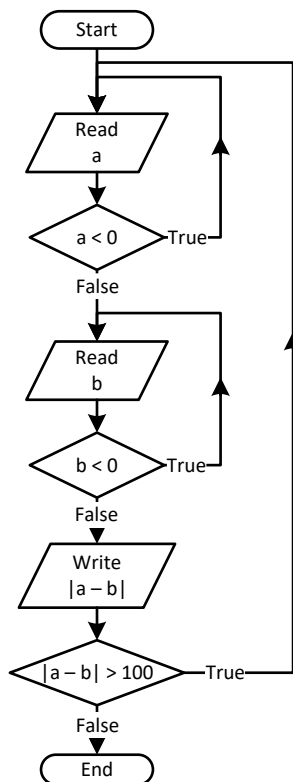
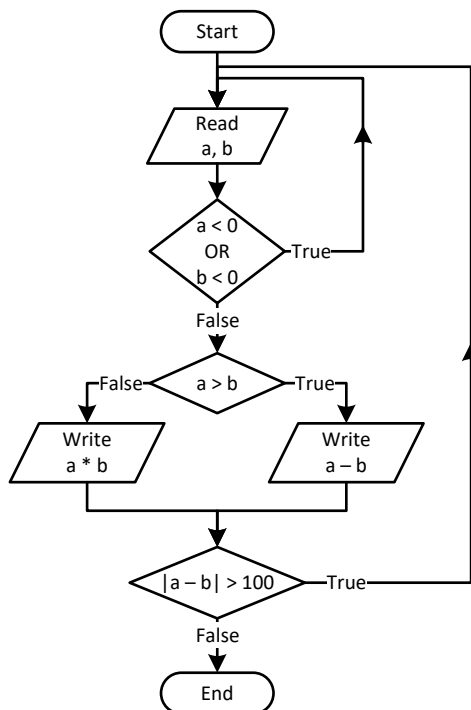


8. Solution



9. Solution

10. Solution**11. Solution**

12. Solution**13. Solution**

14. Solution

```
i = 1;
do {
    Console.WriteLine(i);
    i += 5;
} while (i <= 500);
Console.WriteLine("The End");
```

15. Solution

```
static void Main(string[] args) {
    int a, i;

    i = 0;
    a = Int32.Parse(Console.ReadLine());
    do {
        if (i % 2 != 0) {
            Console.WriteLine(i);
        }
        i += 5;
    } while (i < a);
}
```

16. Solution

```
static void Main(string[] args) {
    int a, b, i;

    a = Int32.Parse(Console.ReadLine());
    while (a != -1) {
        do {
            b = Int32.Parse(Console.ReadLine());
        } while (b <= a);
        for (i = a; i <= b; i++) {
            Console.WriteLine(i);
        }
        a = Int32.Parse(Console.ReadLine());
    }
}
```

17. Solution

```
static void Main(string[] args) {
    int i;
    double P, S, a;

    i = 1;
    S = 0;
    P = 1;
    a = 0;
```

```
while (true) {  
    if (i < 45) {  
        S += a;  
    }  
    else {  
        P *= a;  
    }  
    i++;  
    if (i >= 90) break;  
    a = Double.Parse(Console.ReadLine());  
}  
  
Console.WriteLine(S + " " + P);  
}
```

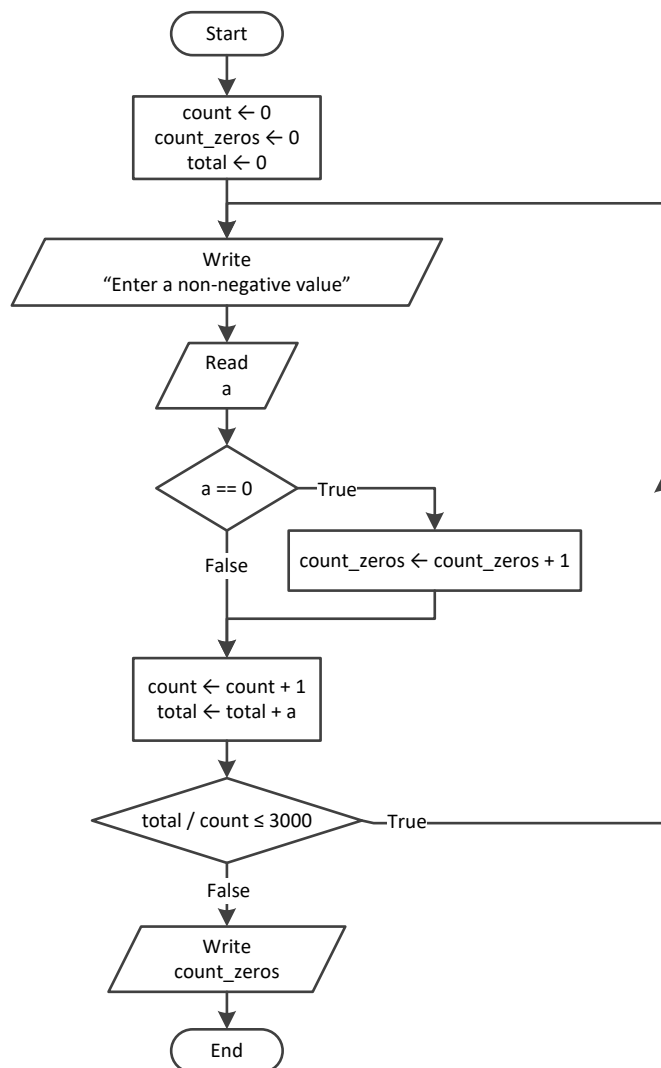
Chapter 30

30.7 Review Questions: True/False

- | | |
|----------|----------|
| 1. true | 6. false |
| 2. false | 7. false |
| 3. false | 8. false |
| 4. true | 9. true |
| 5. false | |

30.8 Review Exercises

1. Solution



```

static void Main(string[] args) {
    int count, count_zeros;
    double a, total;

    count = 0;
    count_zeros = 0;
  
```



```

total = 0;
do {
    Console.Write("Enter a non-negative value: ");
    a = Double.Parse(Console.ReadLine());
    if (a == 0) {
        count_zeros++;
    }
    count++;
    total += a;
} while (total / count <= 3000);
Console.WriteLine(count_zeros);
}

```

2. Solution

First approach

```

static void Main(string[] args) {
    int a, d1, d2, d3, d4, i, r;

    Console.Write("Enter an integer between 1 and 20: ");
    a = Int32.Parse(Console.ReadLine());
    for (i = 1000; i <= 9999; i++) {
        d4 = i % 10;
        r = (int)(i / 10);
        d3 = r % 10;
        r = (int)(r / 10);
        d2 = r % 10;
        d1 = (int)(r / 10);
        if (d1 + d2 + d3 + d4 < a) {
            Console.WriteLine(i);
        }
    }
}

```

Second approach

```

static void Main(string[] args) {
    int a, d1, d2, d3, d4;

    Console.Write("Enter an integer between 1 and 20: ");
    a = Int32.Parse(Console.ReadLine());
    for (d1 = 1; d1 <= 9; d1++) {
        for (d2 = 0; d2 <= 9; d2++) {
            for (d3 = 0; d3 <= 9; d3++) {
                for (d4 = 0; d4 <= 9; d4++) {
                    if (d1 + d2 + d3 + d4 < a) {
                        Console.WriteLine(d1 * 1000 + d2 * 100 + d3 * 10 + d4);
                    }
                }
            }
        }
    }
}

```

3. Solution

First approach

```
static void Main(string[] args) {
    int d1, d2, d3, d4, i, r;

    for (i = 1000; i <= 9999; i++) {
        d4 = i % 10;
        r = (int)(i / 10);
        d3 = r % 10;
        r = (int)(r / 10);
        d2 = r % 10;
        d1 = (int)(r / 10);
        if (d1 > d2 && d2 == d3 && d3 < d4) {
            Console.WriteLine(i);
        }
    }
}
```

Second approach

```
static void Main(string[] args) {
    int d1, d2, d3, d4;

    for (d1 = 1; d1 <= 9; d1++) {
        for (d2 = 0; d2 <= 9; d2++) {
            for (d3 = 0; d3 <= 9; d3++) {
                for (d4 = 0; d4 <= 9; d4++) {
                    if (d1 > d2 && d2 == d3 && d3 < d4) {
                        Console.WriteLine(d1 * 1000 + d2 * 100 + d3 * 10 + d4);
                    }
                }
            }
        }
    }
}
```

4. Solution

First approach

```
static void Main(string[] args) {
    int x, count;

    Console.Write("Enter a number: ");
    x = Int32.Parse(Console.ReadLine());

    count = 0;

    while (x != 0) {
        count++;
        x = (int)(x / 10);
    }

    Console.WriteLine(count);
}
```

```
}
```

Second approach

```
static void Main(string[] args) {
    int x, count;

    Console.Write("Enter a number: ");
    x = Int32.Parse(Console.ReadLine());

    //Convert the absolute value of x to string and get its length
    count = Math.Abs(x).ToString().Length; //Or you can do the following:
                                           //count = (" " + Math.Abs(x)).Length;

    Console.WriteLine(count);
}
```

5. Solution

```
x = Int32.Parse(Console.ReadLine());
while (x != 1 && x != 0) {
    Console.WriteLine("Error");
    x = Int32.Parse(Console.ReadLine());
}
```

6. Solution

```
do {
    gender = Console.ReadLine().ToUpper();
} while (gender != "M" && gender != "F");
```

7. Solution

```
static void Main(string[] args) {
    int count, x;
    double y;

    Console.Write("Enter a non-negative number: ");
    x = Int32.Parse(Console.ReadLine());
    count = 0;
    while (x < 0) {
        count++;
        if (count == 2) break;

        Console.WriteLine("Error: Invalid number!");
        Console.Write("Enter a non-negative number: ");
        x = Int32.Parse(Console.ReadLine());
    }

    if (count < 2) {
        y = Math.Sqrt(x);
        Console.WriteLine(y);
    }
    else {
        Console.WriteLine("Dude, you are dumb!");
    }
}
```

```

    }
}

```

8. Solution

```

static void Main(string[] args) {
    string answer;
    double area, r;

    do {
        Console.Write("Enter the length of a radius: ");
        r = Double.Parse(Console.ReadLine());
        while (r <= 0) {
            Console.Write("Invalid radius. Enter the length of a radius: ");
            r = Double.Parse(Console.ReadLine());
        }

        area = Math.PI * Math.Pow(r, 2);
        Console.WriteLine("The area is: " + area);

        Console.Write("Would you like to repeat? ");
        answer = Console.ReadLine();
    } while (answer.ToUpper() == "YES");
}

```

9. Solution

```

static void Main(string[] args) {
    int x, y;

    for (x = -100; x <= 100; x++) {
        for (y = -100; y <= 100; y++) {
            if (5 * x + 3 * Math.Pow(y, 2) == 0) {
                Console.WriteLine(x + ", " + y);
            }
        }
    }
}

```

10. Solution

```

static void Main(string[] args) {
    int x, y, z;

    for (x = -10; x <= 10; x++) {
        for (y = -10; y <= 10; y++) {
            for (z = -10; z <= 10; z++) {
                if ((x + y) / 2.0 + 3.0 * Math.Pow(z, 2) / (x + 3 * y + 45) == x / 3.0) {
                    Console.WriteLine(x + ", " + y + ", " + z);
                }
            }
        }
    }
}

```

```

    }
}

```

11. Solution

```

static void Main(string[] args) {
    int m1, m2, m3, s;

    m1 = Int32.Parse(Console.ReadLine());
    m2 = Int32.Parse(Console.ReadLine());
    m3 = Int32.Parse(Console.ReadLine());

    s = 0;
    while (m2 != 0) {
        if (m2 % 2 != 0) {
            s += m1;
        }
        m1 *= 2;
        m2 = (int)(m2 / 2);
    }

    m1 = s;
    m2 = m3;

    s = 0;
    while (m2 != 0) {
        if (m2 % 2 != 0) {
            s += m1;
        }
        m1 *= 2;
        m2 = (int)(m2 / 2);
    }

    Console.WriteLine(s);
}

```

12. Solution

```

static void Main(string[] args) {
    int x, number_of_divisors, i;

    x = Int32.Parse(Console.ReadLine());
    while (x <= 0) {
        Console.WriteLine("Error! You must enter a positive integer");
        x = Int32.Parse(Console.ReadLine());
    }

    number_of_divisors = 2;
    for (i = 2; i <= (int)(x / 2); i++) {
        if (x % i == 0) {
            number_of_divisors++;
        }
    }
}

```

```

    Console.WriteLine(number_of_divisors);
}

```

13. Solution

```

static void Main(string[] args) {
    int x, number_of_divisors, i;

    Console.Write("Enter an integer greater than 1: ");
    x = Int32.Parse(Console.ReadLine());
    while (x <= 1) {
        Console.WriteLine("Error!");
        x = Int32.Parse(Console.ReadLine());
    }

    number_of_divisors = 2;
    for (i = 2; i <= (int)(x / 2); i++) {
        if (x % i == 0) {
            number_of_divisors++;
            break;
        }
    }

    if (number_of_divisors == 2) {
        Console.WriteLine("Number " + x + " is prime");
    }
}

```

14. Solution

```

static void Main(string[] args) {
    int a, b, c, i, number_of_divisors, x;

    Console.Write("Enter an integer greater than 1: ");
    a = Int32.Parse(Console.ReadLine());
    while (a < 2) {
        Console.Write("Wrong number. Please enter an integer greater than 1: ");
        a = Int32.Parse(Console.ReadLine());
    }

    Console.Write("Enter a second integer greater than 1: ");
    b = Int32.Parse(Console.ReadLine());
    while (b < 2) {
        Console.Write("Wrong number. Please enter a second integer greater than 1: ");
        b = Int32.Parse(Console.ReadLine());
    }

    if (a > b) {
        c = a;
        a = b;
        b = c;
    }
}

```

```

for (x = a; x <= b; x++) {
    number_of_divisors = 2;
    i = 2;
    while (i <= (int)(x / 2) && number_of_divisors == 2) {
        if (x % i == 0) {
            number_of_divisors++;
        }
        i++;
    }
    if (number_of_divisors == 2) {
        Console.WriteLine("Number " + x + " is prime");
    }
}

```

15. Solution

```

static void Main(string[] args) {
    int a, b, c, d1, d2, d3, d4, r, x;

    Console.Write("Enter a positive four-digit integer: ");
    a = Int32.Parse(Console.ReadLine());
    while (a < 1000 || a > 9999) {
        Console.Write("Wrong number. Please enter a positive four-digit integer: ");
        a = Int32.Parse(Console.ReadLine());
    }

    Console.Write("Enter a second positive four-digit integer: ");
    b = Int32.Parse(Console.ReadLine());
    while (b < 1000 || b > 9999) {
        Console.Write("Wrong number. Please enter a second positive four-digit integer: ");
        b = Int32.Parse(Console.ReadLine());
    }

    if (a > b) {
        c = a;
        a = b;
        b = c;
    }

    for (x = a; x <= b; x++) {
        d4 = x % 10;
        r = (int)(x / 10);
        d3 = r % 10;
        r = (int)(r / 10);
        d2 = r % 10;
        d1 = (int)(r / 10);

        if (d1 == d4 && d2 == d3) {
            Console.WriteLine(x);
        }
    }
}

```

```
}
```

16. Solution

```
static void Main(string[] args) {  
    int i;  
  
    for (i = 0; i <= 30; i++) {  
        Console.WriteLine(Math.Pow(2, i));  
    }  
}
```

17. Solution

```
static void Main(string[] args) {  
    int i, offset;  
  
    offset = 10;  
    i = 1;  
    while (i <= 401) {  
        Console.WriteLine(i);  
        i += offset;  
        offset += 2;  
    }  
}
```

18. Solution

```
static void Main(string[] args) {  
    int i;  
  
    for (i = 1; i <= 100; i++) {  
        Console.WriteLine(-i + "\n" + i);  
    }  
}
```

19. Solution

First approach

```
static void Main(string[] args) {  
    int i, offset, value;  
  
    value = 0;  
    for (i = 1; i <= 8; i++) {  
        offset = (int)Math.Pow(10, i - 1);  
        value += offset;  
        Console.WriteLine(value);  
    }  
}
```

Second approach

```
static void Main(string[] args) {  
    int i;  
    string value;
```



```
value = "1";  
for (i = 1; i <= 8; i++) {  
    Console.WriteLine(value);  
    value += "1";  
}  
}
```

20. Solution

```
static void Main(string[] args) {  
    int a, fib, fib_prev, fib_prev_prev, i;  
  
    a = Int32.Parse(Console.ReadLine());  
  
    fib_prev_prev = 0;  
    fib_prev = 1;  
    fib = 1;  
    for (i = 1; i <= a; i++) {  
        Console.WriteLine(fib);  
        fib = fib_prev + fib_prev_prev;  
        fib_prev_prev = fib_prev;  
        fib_prev = fib;  
    }  
}
```

21. Solution

```
static void Main(string[] args) {  
    int a, fib, fib_prev, fib_prev_prev;  
  
    a = Int32.Parse(Console.ReadLine());  
  
    fib_prev_prev = 0;  
    fib_prev = 1;  
    fib = 1;  
    while (fib < a) {  
        Console.WriteLine(fib);  
        fib = fib_prev + fib_prev_prev;  
        fib_prev_prev = fib_prev;  
        fib_prev = fib;  
    }  
}
```

22. Solution

```
static void Main(string[] args) {  
    int denominator, i, n, nominator;  
    double y;  
  
    Console.Write("Enter a positive integer: ");  
    n = Int32.Parse(Console.ReadLine());  
    while (n <= 0) {
```

```

        Console.Write("Wrong number. Please enter a positive integer: ");
        n = Int32.Parse(Console.ReadLine());
    }

    nominator = 0;
    for (i = 2; i <= 2 * n; i += 2) {
        nominator += i;
    }

    denominator = 1;
    for (i = 1; i <= n; i++) {
        denominator *= i;
    }

    y = nominator / (double)denominator;
    Console.WriteLine(y);
}

```

23. Solution

```

static void Main(string[] args) {
    int i, n, nominator, sign;
    double y;

    Console.Write("Enter a positive integer: ");
    n = Int32.Parse(Console.ReadLine());
    while (n <= 0) {
        Console.Write("Wrong number. Please enter a positive integer: ");
        n = Int32.Parse(Console.ReadLine());
    }

    nominator = 0;
    sign = 1;
    for (i = 1; i <= 2 * n + 1; i += 2) {
        nominator += sign * i;
        sign = -sign;
    }

    y = nominator / (double)n;
    Console.WriteLine(y);
}

```

24. Solution

```

static void Main(string[] args) {
    int i, n, sign;
    double y;

    Console.Write("Enter a positive integer: ");
    n = Int32.Parse(Console.ReadLine());
    while (n <= 0) {
        Console.Write("Wrong number. Please enter a positive integer: ");
        n = Int32.Parse(Console.ReadLine());
    }
}

```

```

    }

    y = 0.5; //This is equal to the first two terms: 1 - 1 / 2

    sign = 1;
    for (i = 3; i <= n; i += 2) {
        y += sign / (double)i;
        sign = -sign;
    }

    Console.WriteLine(y);
}

```

25. Solution

```

static void Main(string[] args) {
    int i, n;
    double y;

    Console.Write("Enter a positive integer: ");
    n = Int32.Parse(Console.ReadLine());
    while (n <= 0) {
        Console.Write("Wrong number. Please enter a positive integer: ");
        n = Int32.Parse(Console.ReadLine());
    }

    y = 0;
    for (i = 1; i <= n; i++) {
        y += 1 / Math.Pow(i, n - i + 1);
    }

    Console.WriteLine(y);
}

```

26. Solution

```


static void Main(string[] args) {
    int factorial, i, n;

    Console.Write("Enter a non-negative integer: ");
    n = Int32.Parse(Console.ReadLine());

    factorial = 1;
    for (i = 1; i <= n; i++) {
        factorial *= i;
    }

    Console.WriteLine(factorial);
}

```

 Please note that this code operates properly for all non-negative integers, including zero.

27. Solution

First approach

```
const double ACCURACY = 0.00001;

static void Main(string[] args) {
    int i, j;
    double factorial, exponential_previous, exponential, x;

    x = Double.Parse(Console.ReadLine());

    exponential = 0;
    i = 0;
    do {
        exponential_previous = exponential;

        factorial = 1;
        for (j = 1; j <= i; j++) {
            factorial *= j;
        }

        exponential += Math.Pow(x, i) / factorial;

        i++;
    } while (Math.Abs(exponential - exponential_previous) > ACCURACY);

    Console.WriteLine("e(" + x + ") ~= " + exponential);
}
```

Second approach

```
const double ACCURACY = 0.00001;

static void Main(string[] args) {
    int i;
    double factorial, exponential_previous, exponential, x;

    x = Double.Parse(Console.ReadLine());

    exponential = 1;
    i = 1;
    factorial = 1;
    do {
        exponential_previous = exponential;

        factorial *= i;

        exponential += Math.Pow(x, i) / factorial;

        i++;
    } while (Math.Abs(exponential - exponential_previous) > ACCURACY);

    Console.WriteLine("e(" + x + ") ~= " + exponential);
}
```

28. Solution

First approach

```
const double ACCURACY = 0.00001;

static void Main(string[] args) {
    int i, j, sign;
    double factorial;
    double sinus, sinus_previous, x;

    x = Double.Parse(Console.ReadLine());

    sign = 1;
    sinus = 0;
    i = 1;
    do {
        sinus_previous = sinus;

        factorial = 1;
        for (j = 1; j <= i; j++) {
            factorial *= j;
        }

        sinus += sign * Math.Pow(x, i) / factorial;

        sign = -sign;
        i += 2;
    } while (Math.Abs(sinus - sinus_previous) > ACCURACY);

    Console.WriteLine("sin(" + x + ") ~= " + sinus);
}
```

Second approach

```
const double ACCURACY = 0.00001;

static void Main(string[] args) {
    int i, sign;
    double factorial;
    double sinus, sinus_previous, x;

    x = Double.Parse(Console.ReadLine());

    sign = -1;
    sinus = x;
    i = 3;
    factorial = 1;
    do {
        sinus_previous = sinus;

        factorial *= i * (i - 1);

        sinus += sign * Math.Pow(x, i) / factorial;

        sign = -sign;
    }
```

```

        i += 2;
    } while (Math.Abs(sinus - sinus_previous) > ACCURACY);

    Console.WriteLine("sin(" + x + ") ~= " + sinus);
}

```

29. Solution

First approach

```

const double ACCURACY = 0.00001;

static void Main(string[] args) {
    int i, j, sign;
    double factorial;
    double cosinus, cosinus_previous, x;

    x = Double.Parse(Console.ReadLine());

    sign = 1;
    cosinus = 0;
    i = 0;
    do {
        cosinus_previous = cosinus;

        factorial = 1;
        for (j = 1; j <= i; j++) {
            factorial *= j;
        }

        cosinus += sign * Math.Pow(x, i) / factorial;

        sign = -sign;
        i += 2;
    } while (Math.Abs(cosinus - cosinus_previous) > ACCURACY);

    Console.WriteLine("cos(" + x + ") ~= " + cosinus);
}

```

Second approach

```

const double ACCURACY = 0.00001;

static void Main(string[] args) {
    int i, sign;
    double factorial;
    double cosinus, cosinus_previous, x;

    x = Double.Parse(Console.ReadLine());

    sign = -1;
    cosinus = 1;
    i = 2;
    factorial = 1;
    do {
        cosinus_previous = cosinus;

```

```

        factorial *= i * (i - 1);

        cosinus += sign * Math.Pow(x, i) / factorial;

        sign = -sign;
        i += 2;
    } while (Math.Abs(cosinus - cosinus_previous) > ACCURACY);

    Console.WriteLine("cos(" + x + ") ~= " + cosinus);
}

```

30. Solution

```

static void Main(string[] args) {
    int i;
    double maximum, total, t;

    maximum = -460;
    total = 0;
    for (i = 1; i <= 31; i++) {
        Console.Write("Enter temperature for day " + i + ": ");
        t = Double.Parse(Console.ReadLine());
        while (t < -459.67) {
            Console.WriteLine("Error! Wrong temperature.");
            Console.Write("Enter temperature for day " + i + ": ");
            t = Double.Parse(Console.ReadLine());
        }

        total += t;
        if (t > maximum) {
            maximum = t;
        }
    }

    Console.WriteLine(total / 31 + " " + maximum);
}

```

31. Solution

```

static void Main(string[] args) {
    int hour, max_hour, max_minutes, min_hour, min_minutes, minutes;
    double level, maximum, minimum;

    level = Double.Parse(Console.ReadLine());
    if (level != 9999) {
        hour = Int32.Parse(Console.ReadLine());
        minutes = Int32.Parse(Console.ReadLine());

        maximum = level;
        max_hour = hour;
        max_minutes = minutes;

        minimum = level;
        min_hour = hour;
    }
}

```

```

    min_minutes = minutes;

    level = Double.Parse(Console.ReadLine());
    while (level != 9999) {
        hour = Int32.Parse(Console.ReadLine());
        minutes = Int32.Parse(Console.ReadLine());

        if (level > maximum) {
            maximum = level;
            max_hour = hour;
            max_minutes = minutes;
        }

        if (level < minimum) {
            minimum = level;
            min_hour = hour;
            min_minutes = minutes;
        }

        level = Double.Parse(Console.ReadLine());
    }

    Console.WriteLine(maximum + ", " + max_hour + ", " + max_minutes);
    Console.WriteLine(minimum + ", " + min_hour + ", " + min_minutes);
}
}

```

32. Solution

```

static void Main(string[] args) {
    int a, b, c, i;
    bool failure;

    string alphabet = "abcdefghijklmnopqrstuvwxyz";

    do {
        Console.Write("Enter an integer between 1 and 26: ");
        a = Int32.Parse(Console.ReadLine());

        failure = false;
        if (a < 1) {
            Console.WriteLine("Please enter positive integers!");
            failure = true;
        }
        else if (a > 26) {
            Console.WriteLine("Please enter a value less than or equal to 26!");
            failure = true;
        }
    } while (failure);

    do {
        Console.Write("Enter an integer between 1 and 26: ");
        b = Int32.Parse(Console.ReadLine());
    }
}

```



```

    failure = false;
    if (b < 1) {
        Console.WriteLine("Please enter positive integers!");
        failure = true;
    }
    else if (b > 26) {
        Console.WriteLine("Please enter a value less than or equal to 26!");
        failure = true;
    }
} while (failure);

if (a > b) {
    c = a;
    a = b;
    b = c;
}

for (i = a; i <= b; i++) {
    Console.Write(alphabet[i - 1]);
}
}

```

33. Solution

```

static void Main(string[] args) {
    int attempts, guess, secret_number;

    Random rnd = new Random();

    secret_number = rnd.Next(1, 101);

    attempts = 1;
    Console.Write("Enter a guess: ");
    guess = Int32.Parse(Console.ReadLine());
    while (guess != secret_number) {
        if (guess > secret_number) {
            Console.WriteLine("Your guess is bigger than my secret number. Try again.");
        }
        else {
            Console.WriteLine("Your guess is smaller than my secret number. Try again.");
        }
        attempts++;
        Console.Write("Enter a guess: ");
        guess = Int32.Parse(Console.ReadLine());
    }
    Console.WriteLine("You found it!");
    Console.WriteLine("Attempts: " + attempts);
}

```

34. Solution

```

static void Main(string[] args) {

```

```

int attempts = 0, first_player_attempts = 0, guess, i, secret_number;

Random rnd = new Random();

for (i = 1; i <= 2; i++) {
    secret_number = rnd.Next(1, 101);

    attempts = 1;
    Console.Write("Enter a guess: ");
    guess = Int32.Parse(Console.ReadLine());
    while (guess != secret_number) {
        if (guess > secret_number) {
            Console.WriteLine("Your guess is bigger than my secret number. Try again.");
        }
        else {
            Console.WriteLine("Your guess is smaller than my secret number. Try again.");
        }
        attempts++;
        Console.Write("Enter a guess: ");
        guess = Int32.Parse(Console.ReadLine());
    }
    Console.WriteLine("You found it!");
    Console.WriteLine("Attempts: " + attempts);

    if (i == 1) {
        first_player_attempts = attempts;
    }
}

if (first_player_attempts < attempts) {
    Console.WriteLine("First player wins!");
}
else if (first_player_attempts > attempts) {
    Console.WriteLine("Second player wins!");
}
else {
    Console.WriteLine("It's a draw");
}
}

```

35. Solution

```

static void Main(string[] args) {
    int choice, diagonal;

    while (true) {
        Console.WriteLine("1. 4/3 TV Screen");
        Console.WriteLine("2. 16/9 TV Screen");
        Console.WriteLine("3. Exit");
        Console.Write("Enter a choice: ");
        choice = Int32.Parse(Console.ReadLine());

        if (choice == 3) {

```

```

        break;
    }
    else if (choice == 1) {
        Console.WriteLine("Enter diagonal: ");
        diagonal = Int32.Parse(Console.ReadLine());
        Console.WriteLine("Width: " + (diagonal * 0.8));
        Console.WriteLine("Height: " + (diagonal * 0.6));
    }
    else if (choice == 2) {
        Console.WriteLine("Enter diagonal: ");
        diagonal = Int32.Parse(Console.ReadLine());
        Console.WriteLine("Width: " + (diagonal * 0.87));
        Console.WriteLine("Height: " + (diagonal * 0.49));
    }
}
}
}

```

36. Solution

```

static void Main(string[] args) {
    int count_a, count_a_boys, count_b, count_cdef_girls, grade;
    int i, maximum, minimum, n, total, total_a, total_a_boys, total_b;
    string gender;

    Console.Write("Enter total number of students: ");
    n = Int32.Parse(Console.ReadLine());
    while (n <= 0) {
        Console.Write("Wrong number. Please enter total number of students: ");
        n = Int32.Parse(Console.ReadLine());
    }

    total = 0;
    total_a = 0;
    count_a = 0;
    total_b = 0;
    count_b = 0;
    total_a_boys = 0;
    count_a_boys = 0;
    count_cdef_girls = 0;

    maximum = -1;
    minimum = 101;

    for (i = 1; i <= n; i++) {
        Console.Write("Enter grade for student No " + i + ": ");
        grade = Int32.Parse(Console.ReadLine());
        while (grade < 0 || grade > 100) {
            Console.Write("Wrong grade. Please enter grade for student No " + i + ": ");
            grade = Int32.Parse(Console.ReadLine());
        }

        Console.Write("Enter gender for student No " + i + ": ");
    }
}

```

```

gender = Console.ReadLine().ToUpper();
while (gender != "M" && gender != "F") {
    Console.WriteLine("Wrong gender. Please enter gender for student No " + i + ": ");
    gender = Console.ReadLine().ToUpper();
}

if (grade >= 90 && grade <= 100) {
    total_a += grade;
    count_a++;
    if (gender == "M") {
        total_a_boys += grade;
        count_a_boys++;
    }
}
else if (grade >= 80 && grade <= 89) {
    total_b += grade;
    count_b++;
}
else {
    if (gender == "F") {
        count_cdef_girls++;
    }
}

if (grade > maximum) {
    maximum = grade;
}

if (grade < minimum) {
    minimum = grade;
}

total += grade;
}

if (count_a > 0) {
    Console.WriteLine("The average value of those who got an 'A' is: ");
    Console.WriteLine(total_a / (double)count_a);
}
if (count_b > 0) {
    Console.WriteLine("The average value of those who got a 'B' is: ");
    Console.WriteLine(total_b / (double)count_b);
}
if (count_a_boys > 0) {
    Console.WriteLine("The average value of boys who got an 'A' is: ");
    Console.WriteLine(total_a_boys / (double)count_a_boys);
}
Console.WriteLine("The total number of girls that got less than 'B' is: " + count_cdef_girls);
Console.WriteLine("The highest grade is: " + maximum);
Console.WriteLine("The lowest grade is: " + minimum);
Console.WriteLine("The average grade of the whole class is: " + total / (double)n);
}

```

37. Solution

```

static void Main(string[] args) {
    double amount, discount;
    string answer;

    do {
        Console.Write("Enter amount: ");
        amount = Double.Parse(Console.ReadLine());
        while (amount <= 0) {
            Console.WriteLine("Wrong amount. Please enter amount: ");
            amount = Double.Parse(Console.ReadLine());
        }

        if (amount < 20) {
            discount = 0;
        }
        else if (amount < 50) {
            discount = 3;
        }
        else if (amount < 100) {
            discount = 5;
        }
        else {
            discount = 10;
        }

        Console.WriteLine("Discount: " + discount + "%");
        Console.WriteLine("Amount to pay (discount included): " + (amount - amount * discount / 100));

        Console.Write("Would you like to repeat? ");
        answer = Console.ReadLine().ToUpper();
    } while (answer == "YES");
}

```

38. Solution

```

const double TAX_RATE = 0.25;

static void Main(string[] args) {
    int kwh;
    double t;

    Console.Write("Enter number of Kilowatt-hours consumed: ");
    kwh = Int32.Parse(Console.ReadLine());
    while (kwh < 0 && kwh != -1) {
        Console.Write("Wrong value. Please enter number of Kilowatt-hours consumed: ");
        kwh = Int32.Parse(Console.ReadLine());
    }

    while (kwh != -1) {
        if (kwh <= 400) {
            t = kwh * 0.11;

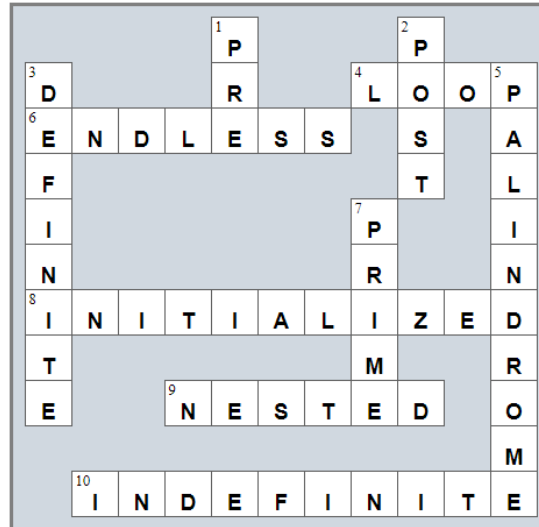
```

```
}  
else if (kwh <= 1500) {  
    t = 400 * 0.11 + (kwh - 400) * 0.22;  
}  
else if (kwh <= 3500) {  
    t = 400 * 0.11 + 1100 * 0.22 + (kwh - 1500) * 0.25;  
}  
else {  
    t = 400 * 0.11 + 1100 * 0.22 + 2000 * 0.25 + (kwh - 3500) * 0.50;  
}  
  
t += t * TAX_RATE;  
Console.WriteLine("Total amount to pay (taxes included): " + t);  
  
Console.Write("Enter number of Kilowatt-hours consumed: ");  
kwh = Int32.Parse(Console.ReadLine());  
while (kwh < 0 && kwh != -1) {  
    Console.Write("Wrong value. Please enter number of Kilowatt-hours consumed: ");  
    kwh = Int32.Parse(Console.ReadLine());  
}  
}  
}
```

Review in “Loop Control Structures”

Review Crossword Puzzle

1.



Chapter 31

31.13 Review Questions: True/False

- | | |
|-----------|-----------|
| 1. true | 21. true |
| 2. true | 22. false |
| 3. false | 23. true |
| 4. false | 24. false |
| 5. false | 25. true |
| 6. true | 26. false |
| 7. false | 27. false |
| 8. true | 28. true |
| 9. false | 29. false |
| 10. true | 30. true |
| 11. true | 31. true |
| 12. true | 32. false |
| 13. false | 33. false |
| 14. false | 34. true |
| 15. false | 35. true |
| 16. true | 36. true |
| 17. false | 37. false |
| 18. true | 38. false |
| 19. true | 39. true |
| 20. false | |

31.14 Review Questions: Multiple Choice

- | | |
|------|-------|
| 1. b | 8. d |
| 2. a | 9. c |
| 3. c | 10. a |
| 4. b | 11. b |
| 5. d | 12. a |
| 6. b | 13. b |
| 7. d | 14. b |

31.15 Review Exercises

1. Solution

Weights =	170	0	} People
	190	1	
	193	2	
	165	3	
	200	4	

2. Solution

Names =	John Thompson	Weights =	170	0	} People
	Chloe Brown		190	1	
	Ryan Miller		193	2	
	Antony Harris		165	3	
	Alexander Lewis		200	4	
	Samantha Clark		170	5	
	Ava Parker		172	6	

3. Solution

Names =

Toba
Issyk Kul
Baikal
Crater
Karakul

Areas =

440	438	437	0
2408	2405	2402	1
12248	12247	12240	2
21	20	18	3
150	145	142	4

Lakes

Months

0

1

2

June

July

August

Solution

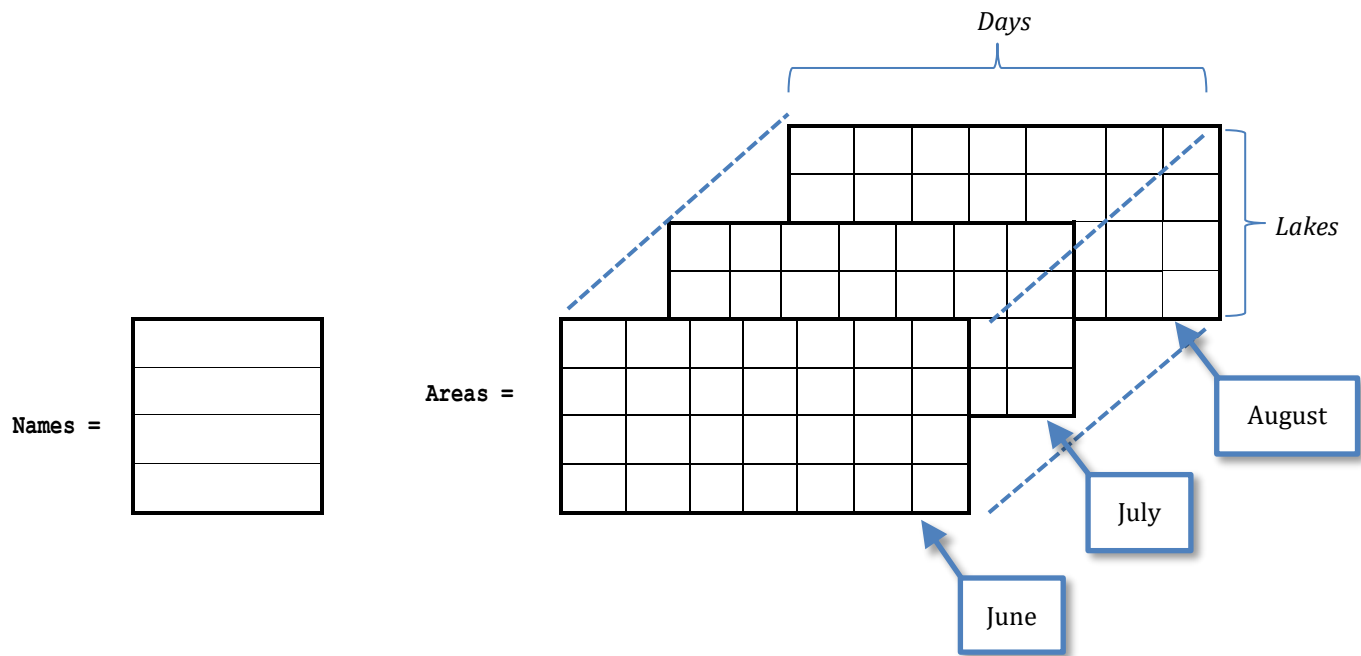
4. Solution

				<i>Dimensions</i>				
				<i>0</i>	<i>1</i>	<i>2</i>		
Boxes =	10	31	15	<i>0</i>				
	15	12	17	<i>1</i>				
	22	10	18	<i>2</i>				
	22	20	12	<i>3</i>				
	26	25	14	<i>4</i>				
	66	26	21	<i>5</i>				
	54	34	24	<i>6</i>				
	64	28	22	<i>7</i>				
	34	12	18	<i>8</i>				
	33	10	10	<i>9</i>				
							<i>Boxes</i>	

5. Solution

Names =	Toba	Areas =	440	Depths =	1660	0	} Lakes
	Issyk Kul		2408		2192	1	
	Baikal		12248		5380	2	
	Crater		21		1950	3	
	Karakul		150		750	4	
	Quesnel		103		2000	5	
	Urmia		2317		52	6	
	Albert		2045		190	7	

6. Solution



7. Solution

Step	Statement	x	a[0]	a[1]	a[2]
1	int[] a = new int[3]	?	?	?	?
2	a[2] = 1	?	?	?	1
3	x = 0	0	?	?	1
4	a[x + a[2]] = 4	0	?	4	1
5	a[x] = a[x + 1] * 4	0	16	4	1

8. Solution

Step	Statement	x	a[0]	a[1]	a[2]	a[3]	a[4]
1	int[] a = new int[5]	?	?	?	?	?	?
2	a[1] = 5	?	?	5	?	?	?
3	x = 0	0	?	5	?	?	?
4	a[x] = 4	0	4	5	?	?	?
5	a[a[0]] = a[x + 1] % 3	0	4	5	?	?	2
6	a[a[0] / 2] = 10	0	4	5	10	?	2
7	x += 2	2	4	5	10	?	2
8	a[x + 1] = a[x] + 9	2	4	5	10	19	2

9. Solution

For input value of 3

Step	Statement	x	a[0]	a[1]	a[2]	a[3]
1	int[] a = new int[4]	?	?	?	?	?
2	a[1] = Int32.Parse(Console.ReadLine())	?	?	3	?	?
3	x = 0	0	?	3	?	?
4	a[x] = 3	0	3	3	?	?
5	a[a[0]] = a[x + 1] % 2	0	3	3	?	1
6	a[a[0] % 2] = 10	0	3	10	?	1
7	x++	1	3	10	?	1
8	a[x + 1] = a[x] + 9	1	3	10	19	1

For input value of 4

Step	Statement	x	a[0]	a[1]	a[2]	a[3]
1	int[] a = new int[4]	?	?	?	?	?
2	a[1] = Int32.Parse(Console.ReadLine())	?	?	4	?	?
3	x = 0	0	?	4	?	?
4	a[x] = 3	0	3	4	?	?
5	a[a[0]] = a[x + 1] % 2	0	3	4	?	0
6	a[a[0] % 2] = 10	0	3	10	?	0
7	x++	1	3	10	?	0
8	a[x + 1] = a[x] + 9	1	3	10	19	0

For input value of 1

Step	Statement	x	a[0]	a[1]	a[2]	a[3]
1	int[] a = new int[4]	?	?	?	?	?
2	a[1] = Int32.Parse(Console.ReadLine())	?	?	1	?	?
3	x = 0	0	?	1	?	?

4	<code>a[x] = 3</code>	0	3	1	?	?
5	<code>a[a[0]] = a[x + 1] % 2</code>	0	3	1	?	3
6	<code>a[a[0] % 2] = 10</code>	0	3	10	?	3
7	<code>x++</code>	1	3	10	?	3
8	<code>a[x + 1] = a[x] + 9</code>	1	3	10	19	3

10. Solution

For input value of 100

Step	Statement	x	a[0]	a[1]	a[2]	a[3]
1	<code>int[] a = new int[4]</code>	?	?	?	?	?
2	<code>a[1] = Int32.Parse(Console.ReadLine())</code>	?	?	100	?	?
3	<code>x = 0</code>	0	?	100	?	?
4	<code>a[x] = 3</code>	0	3	100	?	?
5	<code>a[a[0]] = a[x + 1] % 10</code>	0	3	100	?	0
6	<code>if (a[3] > 5)</code>	false				
7	<code>a[2] = 3</code>	0	3	100	3	0

For input value of 108

Step	Statement	x	a[0]	a[1]	a[2]	a[3]
1	<code>int[] a = new int[4]</code>	?	?	?	?	?
2	<code>a[1] = Int32.Parse(Console.ReadLine())</code>	?	?	108	?	?
3	<code>x = 0</code>	0	?	108	?	?
4	<code>a[x] = 3</code>	0	3	108	?	?
5	<code>a[a[0]] = a[x + 1] % 10</code>	0	3	108	?	8
6	<code>if (a[3] > 5)</code>	true				
7	<code>a[a[0] % 2] = 9</code>	0	3	9	?	8
8	<code>x += 1</code>	1	3	9	?	8
9	<code>a[x + 1] = a[x] + 9</code>	1	3	9	18	8

For input value of 1

Step	Statement	x	a[0]	a[1]	a[2]	a[3]
1	<code>int[] a = new int[4]</code>	?	?	?	?	?
2	<code>a[1] = Int32.Parse(Console.ReadLine())</code>	?	?	1	?	?
3	<code>x = 0</code>	0	?	1	?	?
4	<code>a[x] = 3</code>	0	3	1	?	?
5	<code>a[a[0]] = a[x + 1] % 10</code>	0	3	1	?	1
6	<code>if (a[3] > 5)</code>	false				
7	<code>a[2] = 3</code>	0	3	1	3	1

11. Solution

Step	Statement	x	y	a[0]	a[1]	a[2]
1	int[] a = new int[3]	?	?	?	?	?
2	x = 4	4	?	?	?	?
3	y = x - 1	4	3	?	?	?
4, 5	if (x > y) a[0] = 1; else a[0] = y;	4	3	1	?	?
6	a[1] = x + 3	4	3	1	7	?
7	y = y - 1	4	2	1	7	?
8	a[y] = (x + 5) % 2	4	2	1	7	1

12. Solution

Step	Statement	i	a[0]	a[1]	a[2]	a[3]	a[4]	a[5]
1	int[] a = {17, 12, 45, 12, 12, 49}	?	17	12	45	12	12	49
2	i = 0	0	17	12	45	12	12	49
3	i <= 5	true						
4	if (a[i] == 12)	false						
5	a[i]++	0	18	12	45	12	12	49
6	i++	1	18	12	45	12	12	49
7	i <= 5	true						
8	if (a[i] == 12)	true						
9	a[i]--	1	18	11	45	12	12	49
10	i++	2	18	11	45	12	12	49
11	i <= 5	true						
12	if (a[i] == 12)	false						
13	a[i]++	2	18	11	46	12	12	49
14	i++	3	18	11	46	12	12	49
15	i <= 5	true						
16	if (a[i] == 12)	true						
17	a[i]--	3	18	11	46	11	12	49
18	i++	4	18	11	46	11	12	49
19	i <= 5	true						
20	if (a[i] == 12)	true						
21	a[i]--	4	18	11	46	11	11	49
22	i++	5	18	11	46	11	11	49
23	i <= 5	true						

24	if (a[i] == 12)	false					
25	a[i]++	5	18	11	46	11	50
26	i++	6	18	11	46	11	50
27	i <= 5	false					

13. Solution

Step	Statement	i	a[0]	a[1]	a[2]	a[3]	a[4]	a[5]
1	int[] a = {10, 15, 12, 23, 22, 19}	?	10	15	12	23	22	19
2	i = 1	1	10	15	12	23	22	19
3	i <= 4	true						
4	a[i] = a[i + 1] + a[i - 1]	1	10	22	12	23	22	19
5	i++	2	10	22	12	23	22	19
6	i <= 4	true						
7	a[i] = a[i + 1] + a[i - 1]	2	10	22	45	23	22	19
8	i++	3	10	22	45	23	22	19
9	i <= 4	true						
10	a[i] = a[i + 1] + a[i - 1]	3	10	22	45	67	22	19
11	i++	4	10	22	45	67	22	19
12	i <= 4	true						
13	a[i] = a[i + 1] + a[i - 1]	4	10	22	45	67	86	19
14	i++	5	10	22	45	67	86	19
15	i <= 4	false						

14. Solution

It displays:

Navajo

Cherokee

Sioux

15. Solution

```
const int ELEMENTS = 100;

static void Main(string[] args) {
    int i;

    double[] a = new double[ELEMENTS];
    for (i = 0; i <= ELEMENTS - 1; i++) {
        a[i] = Double.Parse(Console.ReadLine());
    }

    for (i = 0; i <= ELEMENTS - 1; i++) {
        Console.WriteLine(Math.Pow(a[i], 3));
    }
}
```

```
}
```

16. Solution

```
const int ELEMENTS = 80;

static void Main(string[] args) {
    int i;

    double[] a = new double[ELEMENTS];
    for (i = 0; i <= ELEMENTS - 1; i++) {
        a[i] = Double.Parse(Console.ReadLine());
    }

    for (i = 0; i <= ELEMENTS - 1; i++) {
        a[i] = Math.Pow(a[i], 2);
    }

    for (i = ELEMENTS - 1; i >= 0; i--) {
        Console.WriteLine(a[i]);
    }
}
```

17. Solution

```
const int ELEMENTS = 90;

static void Main(string[] args) {
    int i;

    int[] a = new int[ELEMENTS];
    for (i = 0; i <= ELEMENTS - 1; i++) {
        a[i] = Int32.Parse(Console.ReadLine());
    }

    for (i = ELEMENTS - 1; i >= 0; i--) {
        if (a[i] % 5 == 0) {
            Console.WriteLine(a[i]);
        }
    }
}
```

18. Solution

```
const int ELEMENTS = 50;

static void Main(string[] args) {
    int i;

    int[] a = new int[ELEMENTS];
    for (i = 0; i <= ELEMENTS - 1; i++) {
        a[i] = Int32.Parse(Console.ReadLine());
    }

    for (i = 0; i <= ELEMENTS - 1; i++) {
```

```
        if (a[i] % 2 == 0 || a[i] > 10) {  
            Console.WriteLine(a[i]);  
        }  
    }  
}
```

19. Solution

```
const int ELEMENTS = 30;  
  
static void Main(string[] args) {  
    int i;  
    double total;  
  
    double[] a = new double[ELEMENTS];  
    for (i = 0; i <= ELEMENTS - 1; i++) {  
        a[i] = Double.Parse(Console.ReadLine());  
    }  
  
    total = 0;  
    for (i = 0; i <= ELEMENTS - 1; i++) {  
        if (a[i] > 0) {  
            total += a[i];  
        }  
    }  
    Console.WriteLine(total);  
}
```

20. Solution

```
const int ELEMENTS = 50;  
  
static void Main(string[] args) {  
    int i, total;  
  
    int[] a = new int[ELEMENTS];  
    for (i = 0; i <= ELEMENTS - 1; i++) {  
        a[i] = Int32.Parse(Console.ReadLine());  
    }  
  
    total = 0;  
    for (i = 0; i <= ELEMENTS - 1; i++) {  
        if (a[i] >= 10 && a[i] <= 99) {  
            total += a[i];  
        }  
    }  
    Console.WriteLine(total);  
}
```

21. Solution

```
const int ELEMENTS = 40;  
  
static void Main(string[] args) {  
    int i;
```



```

double sum_neg, sum_pos;

double[] a = new double[ELEMENTS];
for (i = 0; i <= ELEMENTS - 1; i++) {
    a[i] = Double.Parse(Console.ReadLine());
}

sum_pos = 0;
sum_neg = 0;
for (i = 0; i <= ELEMENTS - 1; i++) {
    if (a[i] > 0 ) {
        sum_pos += a[i];
    }
    else if (a[i] < 0) {
        sum_neg += a[i];
    }
}
Console.WriteLine(sum_pos + ", " + sum_neg);
}

```

22. Solution

```

const int ELEMENTS = 20;

static void Main(string[] args) {
    int i;
    double total;

    double[] a = new double[ELEMENTS];
    for (i = 0; i <= ELEMENTS - 1; i++) {
        a[i] = Double.Parse(Console.ReadLine());
    }

    total = 0;
    for (i = 0; i <= ELEMENTS - 1; i++) {
        total += a[i];
    }
    Console.WriteLine(total / ELEMENTS);
}

```

23. Solution

```

const int ELEMENTS = 50;

static void Main(string[] args) {
    int i;

    int[] a = new int[ELEMENTS];
    for (i = 0; i <= ELEMENTS - 1; i++) {
        Console.Write("Enter an integer: ");
        a[i] = Int32.Parse(Console.ReadLine());
    }

    for (i = 0; i <= ELEMENTS - 1; i++) {
        if (a[i] < 20) {

```

```
        Console.WriteLine(a[i]);
    }
}
}
```

24. Solution

```
const int ELEMENTS = 60;

static void Main(string[] args) {
    int i;

    double[] a = new double[ELEMENTS];
    for (i = 0; i <= ELEMENTS - 1; i++) {
        Console.Write("Enter a number: ");
        a[i] = Double.Parse(Console.ReadLine());
    }

    for (i = 0; i <= ELEMENTS - 1; i += 2) {
        Console.WriteLine(a[i]);
    }
}
```

25. Solution

```
const int ELEMENTS = 20;

static void Main(string[] args) {
    int i;
    double total;

    double[] a = new double[ELEMENTS];
    for (i = 0; i <= ELEMENTS - 1; i++) {
        Console.Write("Enter a number: ");
        a[i] = Double.Parse(Console.ReadLine());
    }

    total = 0;
    for (i = 0; i <= ELEMENTS - 1; i += 2) {
        total += a[i];
    }
    Console.WriteLine(total);
}
```

26. Solution

```
const int ELEMENTS = 100;

static void Main(string[] args) {
    int i;
    int[] a = new int[ELEMENTS];
    for (i = 0; i <= ELEMENTS - 1; i++) {
        a[i] = i + 1;
    }
    ...
}
```

27. Solution

First approach

```
const int ELEMENTS = 100;

static void Main(string[] args) {
    int i, k;
    int[] a = new int[ELEMENTS];
    k = 2;
    for (i = 0; i <= ELEMENTS - 1; i++) {
        a[i] = k;
        k += 2;
    }
    ...
}
```

Second approach

```
const int ELEMENTS = 100;

static void Main(string[] args) {
    int i;
    int[] a = new int[ELEMENTS];
    for (i = 0; i <= ELEMENTS - 1; i++) {
        a[i] = (i + 1) * 2;
    }
    ...
}
```

28. Solution

```
static void Main(string[] args) {
    int i, n;

    Console.Write("Enter N: ");
    n = Int32.Parse(Console.ReadLine());

    int[] a = new int[n];
    for (i = 1; i <= n; i++) {
        a[i - 1] = (int)Math.Pow(i, 2);
    }

    for (i = 0; i <= n - 1; i++) {
        Console.WriteLine(a[i]);
    }
}
```

29. Solution

```
const int ELEMENTS = 10;

static void Main(string[] args) {
    int i;

    double[] a = new double[ELEMENTS];
    for (i = 1; i <= ELEMENTS - 1; i++) {
        Console.Write("Enter a number: ");
        a[i] = Double.Parse(Console.ReadLine());
    }
}
```

```
    for (i = 0; i <= ELEMENTS - 1; i++) {  
        if (a[i] == (int)a[i]) {  
            Console.WriteLine(i);  
        }  
    }  
}
```

30. Solution

```
const int ELEMENTS = 50;  
  
static void Main(string[] args) {  
    int i, count;  
  
    double[] a = new double[ELEMENTS];  
    for (i = 1; i <= ELEMENTS - 1; i++) {  
        Console.Write("Enter a number: ");  
        a[i] = Double.Parse(Console.ReadLine());  
    }  
  
    count = 0;  
    for (i = 0; i <= ELEMENTS - 1; i++) {  
        if (a[i] < 0) {  
            count++;  
        }  
    }  
    Console.WriteLine(count);  
}
```

31. Solution

```
const int WORDS = 50;  
  
static void Main(string[] args) {  
    int i;  
  
    string[] a = new string[WORDS];  
    for (i = 0; i <= WORDS - 1; i++) {  
        a[i] = Console.ReadLine();  
    }  
  
    for (i = 0; i <= WORDS - 1; i++) {  
        if (a[i].Length >= 10 ) {  
            Console.WriteLine(a[i]);  
        }  
    }  
}
```

32. Solution

```
const int ELEMENTS = 30;  
  
static void Main(string[] args) {  
    int i, k;
```

```

string[] words = new string[ELEMENTS];
for (i = 0; i <= ELEMENTS - 1; i++) {
    words[i] = Console.ReadLine();
}

int[] length_limits = {0, 5, 10, 20};

for (k = 1; k <= 3; k++) {
    for (i = 0; i <= ELEMENTS - 1; i++) {
        if (words[i].Length >= length_limits[k - 1] && words[i].Length < length_limits[k]) {
            Console.WriteLine(words[i]);
        }
    }
}

```

33. Solution

```

const int WORDS = 40;

static void Main(string[] args) {
    int count, i, j;

    string[] a = new string[WORDS];
    for (i = 0; i <= WORDS - 1; i++) {
        Console.Write("Enter a word: ");
        a[i] = Console.ReadLine();
    }

    for (i = 0; i <= WORDS - 1; i++) {
        count = 0;
        for (j = 0; j <= a[i].Length - 1; j++) {
            if (a[i].Substring(j, 1) == "w") { //Alternatively use: if (a[i][j].ToString() == "w")
                count++;
            }
        }
        if (count >= 2) {
            Console.WriteLine(a[i]);
        }
    }
}

```

Chapter 32

32.7 Review Questions: True/False

- | | |
|-----------|-----------|
| 1. false | 15. true |
| 2. true | 16. true |
| 3. false | 17. true |
| 4. false | 18. true |
| 5. false | 19. false |
| 6. true | 20. true |
| 7. false | 21. true |
| 8. true | 22. true |
| 9. true | 23. false |
| 10. true | 24. true |
| 11. true | 25. true |
| 12. true | 26. true |
| 13. false | 27. false |
| 14. true | |

32.8 Review Questions: Multiple Choice

1. b
2. b
3. c
4. a
5. d
6. a
7. d
8. c
9. c
10. c
11. b

32.9 Review Exercises

1. Solution

Step	Statement	x	a						
1	int[,] a = new int[2, 3]	?	<table><tr><td>?</td><td>?</td><td>?</td></tr><tr><td>?</td><td>?</td><td>?</td></tr></table>	?	?	?	?	?	?
?	?	?							
?	?	?							
2	a[0, 2] = 1	?	<table><tr><td>?</td><td>?</td><td>1</td></tr><tr><td>?</td><td>?</td><td>?</td></tr></table>	?	?	1	?	?	?
?	?	1							
?	?	?							
3	x = 0	0	<table><tr><td>?</td><td>?</td><td>1</td></tr><tr><td>?</td><td>?</td><td>?</td></tr></table>	?	?	1	?	?	?
?	?	1							
?	?	?							
4	a[0, x] = 9	0	<table><tr><td>9</td><td>?</td><td>1</td></tr><tr><td>?</td><td>?</td><td>?</td></tr></table>	9	?	1	?	?	?
9	?	1							
?	?	?							
5	a[0, x + a[0, 2]] = 4	0	<table><tr><td>9</td><td>4</td><td>1</td></tr><tr><td>?</td><td>?</td><td>?</td></tr></table>	9	4	1	?	?	?
9	4	1							
?	?	?							
6	a[a[0, 2], 2] = 19	0	<table><tr><td>9</td><td>4</td><td>1</td></tr><tr><td>?</td><td>?</td><td>19</td></tr></table>	9	4	1	?	?	19
9	4	1							
?	?	19							
7	a[a[0, 2], x + 1] = 13	0	<table><tr><td>9</td><td>4</td><td>1</td></tr><tr><td>?</td><td>13</td><td>19</td></tr></table>	9	4	1	?	13	19
9	4	1							
?	13	19							
8	a[a[0, 2], x] = 15	0	<table><tr><td>9</td><td>4</td><td>1</td></tr><tr><td>15</td><td>13</td><td>19</td></tr></table>	9	4	1	15	13	19
9	4	1							
15	13	19							

2. Solution

Step	Statement	i	j	a		
1	int[,] a = new int[2, 3]	?	?	?	?	?
				?	?	?
2	i = 0	0	?	?	?	?
				?	?	?
3	i <= 1	true				
4	j = 0	0	0	?	?	?
				?	?	?

5	j <= 2			true						
6	a[i, j] = (i + 1) * 5 + j	0	0	<table><tr><td>5</td><td>?</td><td>?</td></tr><tr><td>?</td><td>?</td><td>?</td></tr></table>	5	?	?	?	?	?
5	?	?								
?	?	?								
7	j++	0	1	<table><tr><td>5</td><td>?</td><td>?</td></tr><tr><td>?</td><td>?</td><td>?</td></tr></table>	5	?	?	?	?	?
5	?	?								
?	?	?								
8	j <= 2	true								
9	a[i, j] = (i + 1) * 5 + j	0	1	<table><tr><td>5</td><td>6</td><td>?</td></tr><tr><td>?</td><td>?</td><td>?</td></tr></table>	5	6	?	?	?	?
5	6	?								
?	?	?								
10	j++	0	2	<table><tr><td>5</td><td>6</td><td>?</td></tr><tr><td>?</td><td>?</td><td>?</td></tr></table>	5	6	?	?	?	?
5	6	?								
?	?	?								
11	j <= 2	true								
12	a[i, j] = (i + 1) * 5 + j	0	2	<table><tr><td>5</td><td>6</td><td>7</td></tr><tr><td>?</td><td>?</td><td>?</td></tr></table>	5	6	7	?	?	?
5	6	7								
?	?	?								
13	j++	0	3	<table><tr><td>5</td><td>6</td><td>7</td></tr><tr><td>?</td><td>?</td><td>?</td></tr></table>	5	6	7	?	?	?
5	6	7								
?	?	?								
14	j <= 2	false								
15	i++	1	3	<table><tr><td>5</td><td>6</td><td>7</td></tr><tr><td>?</td><td>?</td><td>?</td></tr></table>	5	6	7	?	?	?
5	6	7								
?	?	?								
16	i <= 1	true								
17	j = 0	1	0	<table><tr><td>5</td><td>6</td><td>7</td></tr><tr><td>?</td><td>?</td><td>?</td></tr></table>	5	6	7	?	?	?
5	6	7								
?	?	?								
18	j <= 2	true								
19	a[i, j] = (i + 1) * 5 + j	1	0	<table><tr><td>5</td><td>6</td><td>7</td></tr><tr><td>10</td><td>?</td><td>?</td></tr></table>	5	6	7	10	?	?
5	6	7								
10	?	?								
20	j++	1	1	<table><tr><td>5</td><td>6</td><td>7</td></tr><tr><td>10</td><td>?</td><td>?</td></tr></table>	5	6	7	10	?	?
5	6	7								
10	?	?								
21	j <= 2	true								
22	a[i, j] = (i + 1) * 5 + j	1	1	<table><tr><td>5</td><td>6</td><td>7</td></tr><tr><td>10</td><td>11</td><td>?</td></tr></table>	5	6	7	10	11	?
5	6	7								
10	11	?								
23	j++	1	2	<table><tr><td>5</td><td>6</td><td>7</td></tr><tr><td>10</td><td>11</td><td>?</td></tr></table>	5	6	7	10	11	?
5	6	7								
10	11	?								

24	j <= 2	true									
25	a[i, j] = (i + 1) * 5 + j	1	2	<table><tr><td>5</td><td>6</td><td>7</td></tr><tr><td>10</td><td>11</td><td>12</td></tr></table>		5	6	7	10	11	12
5	6	7									
10	11	12									
26	j++	1	3	<table><tr><td>5</td><td>6</td><td>7</td></tr><tr><td>10</td><td>11</td><td>12</td></tr></table>		5	6	7	10	11	12
5	6	7									
10	11	12									
27	j <= 2	false									
28	i++	2	3	<table><tr><td>5</td><td>6</td><td>7</td></tr><tr><td>10</td><td>11</td><td>12</td></tr></table>		5	6	7	10	11	12
5	6	7									
10	11	12									
29	i <= 1	false									

3. Solution

Step	Statement	i	j	a		
1	int[,] a = new int[3, 3]	?	?	?	?	?
				?	?	?
				?	?	?
2	j = 0	?	0	?	?	?
				?	?	?
				?	?	?
3	j <= 2	True				
4	i = 0	0	0	?	?	?
				?	?	?
				?	?	?
5	i <= 2	True				
6	a[i, j] = (i + 1) * 2 + j * 4	0	0	2	?	?
				?	?	?
				?	?	?
7	i++	1	0	2	?	?
				?	?	?
				?	?	?
8	i <= 2	True				
9	a[i, j] = (i + 1) * 2 + j * 4	1	0	2	?	?
				4	?	?
				?	?	?

10	i++	2	0	<table><tr><td>2</td><td>?</td><td>?</td></tr><tr><td>4</td><td>?</td><td>?</td></tr><tr><td>?</td><td>?</td><td>?</td></tr></table>	2	?	?	4	?	?	?	?	?
2	?	?											
4	?	?											
?	?	?											
11	i <= 2	True											
12	a[i, j] = (i + 1) * 2 + j * 4	2	0	<table><tr><td>2</td><td>?</td><td>?</td></tr><tr><td>4</td><td>?</td><td>?</td></tr><tr><td>6</td><td>?</td><td>?</td></tr></table>	2	?	?	4	?	?	6	?	?
2	?	?											
4	?	?											
6	?	?											
13	i++	3	0	<table><tr><td>2</td><td>?</td><td>?</td></tr><tr><td>4</td><td>?</td><td>?</td></tr><tr><td>6</td><td>?</td><td>?</td></tr></table>	2	?	?	4	?	?	6	?	?
2	?	?											
4	?	?											
6	?	?											
14	i <= 2	False											
15	j++	3	1	<table><tr><td>2</td><td>?</td><td>?</td></tr><tr><td>4</td><td>?</td><td>?</td></tr><tr><td>6</td><td>?</td><td>?</td></tr></table>	2	?	?	4	?	?	6	?	?
2	?	?											
4	?	?											
6	?	?											
16	j <= 2	True											
17	i = 0	0	1	<table><tr><td>2</td><td>?</td><td>?</td></tr><tr><td>4</td><td>?</td><td>?</td></tr><tr><td>6</td><td>?</td><td>?</td></tr></table>	2	?	?	4	?	?	6	?	?
2	?	?											
4	?	?											
6	?	?											
18	i <= 2	True											
19	a[i, j] = (i + 1) * 2 + j * 4	0	1	<table><tr><td>2</td><td>6</td><td>?</td></tr><tr><td>4</td><td>?</td><td>?</td></tr><tr><td>6</td><td>?</td><td>?</td></tr></table>	2	6	?	4	?	?	6	?	?
2	6	?											
4	?	?											
6	?	?											
20	i++	1	1	<table><tr><td>2</td><td>6</td><td>?</td></tr><tr><td>4</td><td>?</td><td>?</td></tr><tr><td>6</td><td>?</td><td>?</td></tr></table>	2	6	?	4	?	?	6	?	?
2	6	?											
4	?	?											
6	?	?											
21	i <= 2	True											
22	a[i, j] = (i + 1) * 2 + j * 4	1	1	<table><tr><td>2</td><td>6</td><td>?</td></tr><tr><td>4</td><td>8</td><td>?</td></tr><tr><td>6</td><td>?</td><td>?</td></tr></table>	2	6	?	4	8	?	6	?	?
2	6	?											
4	8	?											
6	?	?											
23	i++	2	1	<table><tr><td>2</td><td>6</td><td>?</td></tr><tr><td>4</td><td>8</td><td>?</td></tr><tr><td>6</td><td>?</td><td>?</td></tr></table>	2	6	?	4	8	?	6	?	?
2	6	?											
4	8	?											
6	?	?											
24	i <= 2	True											
25	a[i, j] = (i + 1) * 2 + j * 4	2	1	<table><tr><td>2</td><td>6</td><td>?</td></tr><tr><td>4</td><td>8</td><td>?</td></tr><tr><td>6</td><td>10</td><td>?</td></tr></table>	2	6	?	4	8	?	6	10	?
2	6	?											
4	8	?											
6	10	?											

26	i++	3	1	<table><tr><td>2</td><td>6</td><td>?</td></tr><tr><td>4</td><td>8</td><td>?</td></tr><tr><td>6</td><td>10</td><td>?</td></tr></table>	2	6	?	4	8	?	6	10	?
2	6	?											
4	8	?											
6	10	?											
27	i <= 2	False											
28	j++	3	2	<table><tr><td>2</td><td>6</td><td>?</td></tr><tr><td>4</td><td>8</td><td>?</td></tr><tr><td>6</td><td>10</td><td>?</td></tr></table>	2	6	?	4	8	?	6	10	?
2	6	?											
4	8	?											
6	10	?											
29	j <= 2	True											
30	i = 0	0	2	<table><tr><td>2</td><td>6</td><td>?</td></tr><tr><td>4</td><td>8</td><td>?</td></tr><tr><td>6</td><td>10</td><td>?</td></tr></table>	2	6	?	4	8	?	6	10	?
2	6	?											
4	8	?											
6	10	?											
31	i <= 2	True											
32	a[i, j] = (i + 1) * 2 + j * 4	0	2	<table><tr><td>2</td><td>6</td><td>10</td></tr><tr><td>4</td><td>8</td><td>?</td></tr><tr><td>6</td><td>10</td><td>?</td></tr></table>	2	6	10	4	8	?	6	10	?
2	6	10											
4	8	?											
6	10	?											
33	i++	1	2	<table><tr><td>2</td><td>6</td><td>10</td></tr><tr><td>4</td><td>8</td><td>?</td></tr><tr><td>6</td><td>10</td><td>?</td></tr></table>	2	6	10	4	8	?	6	10	?
2	6	10											
4	8	?											
6	10	?											
34	i <= 2	True											
35	a[i, j] = (i + 1) * 2 + j * 4	1	2	<table><tr><td>2</td><td>6</td><td>10</td></tr><tr><td>4</td><td>8</td><td>12</td></tr><tr><td>6</td><td>10</td><td>?</td></tr></table>	2	6	10	4	8	12	6	10	?
2	6	10											
4	8	12											
6	10	?											
36	i++	2	2	<table><tr><td>2</td><td>6</td><td>10</td></tr><tr><td>4</td><td>8</td><td>12</td></tr><tr><td>6</td><td>10</td><td>?</td></tr></table>	2	6	10	4	8	12	6	10	?
2	6	10											
4	8	12											
6	10	?											
37	i <= 2	True											
38	a[i, j] = (i + 1) * 2 + j * 4	2	2	<table><tr><td>2</td><td>6</td><td>10</td></tr><tr><td>4</td><td>8</td><td>12</td></tr><tr><td>6</td><td>10</td><td>14</td></tr></table>	2	6	10	4	8	12	6	10	14
2	6	10											
4	8	12											
6	10	14											
39	i++	3	2	<table><tr><td>2</td><td>6</td><td>10</td></tr><tr><td>4</td><td>8</td><td>12</td></tr><tr><td>6</td><td>10</td><td>14</td></tr></table>	2	6	10	4	8	12	6	10	14
2	6	10											
4	8	12											
6	10	14											
40	i <= 2	False											
41	j++	3	3	<table><tr><td>2</td><td>6</td><td>10</td></tr><tr><td>4</td><td>8</td><td>12</td></tr><tr><td>6</td><td>10</td><td>14</td></tr></table>	2	6	10	4	8	12	6	10	14
2	6	10											
4	8	12											
6	10	14											

42 j <= 2

False

4. Solution

For input value of 5

0	5	10
0	6	12

For input value of 9

0	9	18
0	10	20

For input value of 3

0	3	6
0	4	8

5. Solution

For input value of 13

0	3	3
0	17	18

For input value of 10

0	10	3
0	11	15

For input value of 8

3	3	3
11	12	13

6. Solution

19	5	31
28	6	20

7. Solution

26	29
37	34
59	49

8. Solution

- i. -1 15 22 25 12 16 7 9 1
- ii. 7 9 1 25 12 16 -1 15 22
- iii. 22 15 -1 16 12 25 1 9 7

iv. 1 9 7 16 12 25 22 15 -1
 v. -1 25 7 15 12 9 22 16 1
 vi. 7 25 -1 9 12 15 1 16 22
 vii. 22 16 1 15 12 9 -1 25 7
 viii. 1 16 22 9 12 15 7 25 -1

9. Solution

```
const int ROWS = 10;
const int COLUMNS = 15;

static void Main(string[] args) {
    int i, j;

    int[,] a = new int[ROWS, COLUMNS];
    for (i = 0; i <= ROWS - 1; i++) {
        for (j = 0; j <= COLUMNS - 1; j++) {
            a[i, j] = Int32.Parse(Console.ReadLine());
        }
    }

    for (i = 0; i <= ROWS - 1; i++) {
        for (j = 0; j <= COLUMNS - 1; j++) {
            if (a[i, j] % 2 != 0) {
                Console.WriteLine(i + ", " + j);
            }
        }
    }
}
```

10. Solution

```
const int ROWS = 10;
const int COLUMNS = 6;

static void Main(string[] args) {
    int i, j;

    double[,] a = new double[ROWS, COLUMNS];
    for (i = 0; i <= ROWS - 1; i++) {
        for (j = 0; j <= COLUMNS - 1; j++) {
            a[i, j] = Double.Parse(Console.ReadLine());
        }
    }

    for (i = 0; i <= ROWS - 1; i++) {
        for (j = 0; j <= COLUMNS - 1; j += 2) {
            Console.WriteLine(a[i, j]);
        }
    }
}
```

11. Solution

```
const int ROWS = 12;
const int COLUMNS = 8;

static void Main(string[] args) {
    int i, j;
    double total;

    double[,] a = new double[ROWS, COLUMNS];
    for (i = 0; i <= ROWS - 1; i++) {
        for (j = 0; j <= COLUMNS - 1; j++) {
            a[i, j] = Double.Parse(Console.ReadLine());
        }
    }

    total = 0;
    for (i = 1; i <= ROWS - 1; i += 2) {
        for (j = 0; j <= COLUMNS - 1; j += 2) {
            total += a[i, j];
        }
    }
    Console.WriteLine(total);
}
```

12. Solution

```
const int N = 8;

static void Main(string[] args) {
    int i, j, k;
    double sum_antidiagonal, sum_diagonal;

    double[,] a = new double[N, N];
    for (i = 0; i <= N - 1; i++) {
        for (j = 0; j <= N - 1; j++) {
            a[i, j] = Double.Parse(Console.ReadLine());
        }
    }

    sum_diagonal = 0;
    sum_antidiagonal = 0;
    for (k = 0; k <= N - 1; k++) {
        sum_diagonal += a[k, k];
        sum_antidiagonal += a[k, N - k - 1];
    }
    Console.WriteLine(sum_diagonal / N + ", " + sum_antidiagonal / N);
}
```

13. Solution

```
const int N = 5;
```

```
static void Main(string[] args) {
    int i, j;

    int[,] a = new int[N, N];
    for (i = 0; i <= N - 1; i++) {
        for (j = 0; j <= N - 1; j++) {
            if (i == N - j - 1) {
                a[i, j] = 5;
            }
            else if (i > N - j - 1) {
                a[i, j] = 88;
            }
            else {
                a[i, j] = 11;
            }
        }
    }

    for (i = 0; i <= N - 1; i++) {
        for (j = 0; j <= N - 1; j++) {
            Console.Write(a[i, j] + "\t");
        }
        Console.WriteLine();
    }
}
```

14. Solution

```
const int N = 5;

static void Main(string[] args) {
    int i, j;

    int[,] a = new int[N, N];
    for (i = 0; i <= N - 1; i++) {
        for (j = 0; j <= N - 1; j++) {
            if (i == N - j - 1) {
                a[i, j] = 5;
            }
            else if (i > N - j - 1) {
                a[i, j] = 88;
            }
            else {
                a[i, j] = 11;
            }
            if (i == j) {
                a[i, j] = 0;
            }
        }
    }

    for (i = 0; i <= N - 1; i++) {
```

```
        for (j = 0; j <= N - 1; j++) {
            Console.Write(a[i, j] + "\t");
        }
        Console.WriteLine();
    }
}
```

15. Solution

```
const int ROWS = 5;
const int COLUMNS = 4;

static void Main(string[] args) {
    int i, j;

    double[,] a = new double[ROWS, COLUMNS];
    for (i = 0; i <= ROWS - 1; i++) {
        for (j = 0; j <= COLUMNS - 1; j++) {
            a[i, j] = Double.Parse(Console.ReadLine());
        }
    }

    for (i = 0; i <= ROWS - 1; i++) {
        for (j = 0; j <= COLUMNS - 1; j++) {
            if (a[i, j] == (int)(a[i, j])) {
                Console.WriteLine(i + ", " + j);
            }
        }
    }
}
```

16. Solution

```
const int ROWS = 10;
const int COLUMNS = 4;

static void Main(string[] args) {
    int count, i, j;

    double[,] a = new double[ROWS, COLUMNS];
    for (i = 0; i <= ROWS - 1; i++) {
        for (j = 0; j <= COLUMNS - 1; j++) {
            a[i, j] = Double.Parse(Console.ReadLine());
        }
    }

    count = 0;
    for (i = 0; i <= ROWS - 1; i++) {
        for (j = 0; j <= COLUMNS - 1; j++) {
            if (a[i, j] < 0) {
                count++;
            }
        }
    }
}
```



```
    }  
    Console.WriteLine(count);  
}
```

17. Solution

```
const int ROWS = 3;  
const int COLUMNS = 4;  
  
static void Main(string[] args) {  
    int i, j;  
  
    string[,] a = new string[ROWS, COLUMNS];  
    for (i = 0; i <= ROWS - 1; i++) {  
        for (j = 0; j <= COLUMNS - 1; j++) {  
            a[i, j] = Console.ReadLine();  
        }  
    }  
  
    for (i = 0; i <= ROWS - 1; i++) {  
        for (j = 0; j <= COLUMNS - 1; j++) {  
            Console.Write(a[i, j] + " ");  
        }  
    }  
}
```

18. Solution

```
const int ROWS = 20;  
const int COLUMNS = 14;  
  
static void Main(string[] args) {  
    int i, j;  
  
    string[,] a = new string[ROWS, COLUMNS];  
    for (i = 0; i <= ROWS - 1; i++) {  
        for (j = 0; j <= COLUMNS - 1; j++) {  
            a[i, j] = Console.ReadLine();  
        }  
    }  
  
    for (i = 0; i <= ROWS - 1; i++) {  
        for (j = 0; j <= COLUMNS - 1; j++) {  
            if (a[i, j].Length < 5) {  
                Console.WriteLine(a[i, j]);  
            }  
        }  
    }  
}
```

19. Solution

First approach

```
const int ROWS = 20;
const int COLUMNS = 14;

static void Main(string[] args) {
    int i, j, k;

    string[,] a = new string[ROWS, COLUMNS];
    for (i = 0; i <= ROWS - 1; i++) {
        for (j = 0; j <= COLUMNS - 1; j++) {
            a[i, j] = Console.ReadLine();
        }
    }

    int[] length_limits = {5, 10, 20};

    for (k = 0; k <= 2; k++) {
        for (i = 0; i <= ROWS - 1; i++) {
            for (j = 0; j <= COLUMNS - 1; j++) {
                if (a[i, j].Length < length_limits[k]) {
                    Console.WriteLine(a[i, j]);
                }
            }
        }
    }
}
```

Second approach

```
const int ROWS = 20;
const int COLUMNS = 14;

static void Main(string[] args) {
    int i, j, k;

    string[,] a = new string[ROWS, COLUMNS];
    for (i = 0; i <= ROWS - 1; i++) {
        for (j = 0; j <= COLUMNS - 1; j++) {
            a[i, j] = Console.ReadLine();
        }
    }

    for (k = 0; k <= 2; k++) {
        for (i = 0; i <= ROWS - 1; i++) {
            for (j = 0; j <= COLUMNS - 1; j++) {
                if (a[i, j].Length < 5 * Math.Pow(2, k)) {
                    Console.WriteLine(a[i, j]);
                }
            }
        }
    }
}
```

Chapter 33

33.7 Review Questions: True/False

- | | |
|----------|-----------|
| 1. true | 7. true |
| 2. false | 8. true |
| 3. true | 9. false |
| 4. false | 10. false |
| 5. false | 11. true |
| 6. false | |

33.8 Review Questions: Multiple Choice

- | | |
|------|------|
| 1. a | 4. b |
| 2. b | 5. a |
| 3. c | 6. a |

33.9 Review Exercises

1. Solution

```
const int STUDENTS = 15;
const int TESTS = 5;

static void Main(string[] args) {
    int i, j;

    int[,] grades = new int[STUDENTS, TESTS];
    for (i = 0; i <= STUDENTS - 1; i++) {
        for (j = 0; j <= TESTS - 1; j++) {
            grades[i, j] = Int32.Parse(Console.ReadLine());
        }
    }

    double[] average = new double[STUDENTS];
    for (i = 0; i <= STUDENTS - 1; i++) {
        average[i] = 0;
        for (j = 0; j <= TESTS - 1; j++) {
            average[i] += grades[i, j];
        }
        average[i] /= TESTS;
    }

    for (i = 0; i <= STUDENTS - 1; i++) {
        Console.WriteLine("Student No " + (i + 1) + ": ");

        if (average[i] < 60) {
            Console.WriteLine("E/F");
        }
        else if (average[i] < 70) {
            Console.WriteLine("D");
        }
        else if (average[i] < 80) {
```

```

        Console.WriteLine("C");
    }
    else if (average[i] < 90) {
        Console.WriteLine("B");
    }
    else {
        Console.WriteLine("A");
    }
}
}

```

2. Solution

```

const int OBJECTS = 5;
const int FALLS = 10;

static void Main(string[] args) {
    int i, j, total;

    int[,] g = new int[OBJECTS, FALLS];
    for (i = 0; i <= OBJECTS - 1; i++) {
        for (j = 0; j <= FALLS - 1; j++) {
            g[i, j] = Int32.Parse(Console.ReadLine());
        }
    }

    for (i = 0; i <= OBJECTS - 1; i++) {
        total = 0;
        for (j = 0; j <= FALLS - 1; j++) {
            total += g[i, j];
        }
        Console.WriteLine("Average g for object No " + (i + 1) + ": " + (total / (double)FALLS));
    }

    for (j = 0; j <= FALLS - 1; j++) {
        total = 0;
        for (i = 0; i <= OBJECTS - 1; i++) {
            total += g[i, j];
        }
        Console.WriteLine("Average g for fall No " + (j + 1) + ": " + (total / (double)OBJECTS));
    }

    total = 0;
    for (i = 0; i <= OBJECTS - 1; i++) {
        for (j = 0; j <= FALLS - 1; j++) {
            total += g[i, j];
        }
    }
    Console.WriteLine("Overall average g: " + (total / (double)(OBJECTS * FALLS)));
}

```

3. Solution

```

const int PLAYERS = 15;
const int MATCHES = 12;

static void Main(string[] args) {
    int i, j, total;

    int[,] points = new int[PLAYERS, MATCHES];
    for (i = 0; i <= PLAYERS - 1; i++) {
        for (j = 0; j <= MATCHES - 1; j++) {
            points[i, j] = Int32.Parse(Console.ReadLine());
        }
    }

    for (i = 0; i <= PLAYERS - 1; i++) {
        total = 0;
        for (j = 0; j <= MATCHES - 1; j++) {
            total += points[i, j];
        }
        Console.WriteLine("Total number of points for player No " + (i + 1) + ": " + total);
    }

    for (j = 0; j <= MATCHES - 1; j++) {
        total = 0;
        for (i = 0; i <= PLAYERS - 1; i++) {
            total += points[i, j];
        }
        Console.WriteLine("Total number of points for match No " + (j + 1) + ": " + total);
    }
}

```

4. Solution

```

const int CITIES = 20;
const int HOURS = 24;

static void Main(string[] args) {
    int i, j;
    double total;

    double[,] temperatures = new double[CITIES, HOURS];
    for (i = 0; i <= CITIES - 1; i++) {
        for (j = 0; j <= HOURS - 1; j++) {
            temperatures[i, j] = Int32.Parse(Console.ReadLine());
        }
    }

    for (j = 0; j <= HOURS - 1; j++) {
        total = 0;
        for (i = 0; i <= CITIES - 1; i++) {
            total += temperatures[i, j];
        }
    }
}

```

```

        if (total / CITIES < 10) {
            Console.WriteLine("Hour: " + (j + 1));
        }
    }
}

```

5. Solution

```

const int PLAYERS = 24;
const int MATCHES = 10;

static void Main(string[] args) {
    int i, j, total;

    string[] names = new string[PLAYERS];
    int[,] goals = new int[PLAYERS, MATCHES];
    for (i = 0; i <= PLAYERS - 1; i++) {
        names[i] = Console.ReadLine();
        for (j = 0; j <= MATCHES - 1; j++) {
            goals[i, j] = Int32.Parse(Console.ReadLine());
        }
    }

    for (i = 0; i <= PLAYERS - 1; i++) {
        total = 0;
        for (j = 0; j <= MATCHES - 1; j++) {
            total += goals[i, j];
        }
        Console.WriteLine(names[i] + ": " + (total / (double)MATCHES));
    }

    for (j = 0; j <= MATCHES - 1; j++) {
        total = 0;
        for (i = 0; i <= PLAYERS - 1; i++) {
            total += goals[i, j];
        }
        Console.WriteLine("Match No " + (j + 1) + ": " + total);
    }
}

```

6. Solution

```

const int STUDENTS = 12;
const int LESSONS = 6;

static void Main(string[] args) {
    int i, j, total;

    string[] names = new string[STUDENTS];
    int[,] grades = new int[STUDENTS, LESSONS];
    for (i = 0; i <= STUDENTS - 1; i++) {
        names[i] = Console.ReadLine();
        for (j = 0; j <= LESSONS - 1; j++) {

```

```

        grades[i, j] = Int32.Parse(Console.ReadLine());
    }
}

double[] average = new double[STUDENTS];
for (i = 0; i <= STUDENTS - 1; i++) {
    total = 0;
    for (j = 0; j <= LESSONS - 1; j++) {
        total += grades[i, j];
    }
    average[i] = total / (double)LESSONS;
    Console.WriteLine(names[i] + ": " + average[i]);
}

for (j = 0; j <= LESSONS - 1; j++) {
    total = 0;
    for (i = 0; i <= STUDENTS - 1; i++) {
        total += grades[i, j];
    }
    Console.WriteLine(total / (double)STUDENTS);
}

for (i = 0; i <= STUDENTS - 1; i++) {
    if (average[i] < 60) {
        Console.WriteLine(names[i]);
    }
}

for (i = 0; i <= STUDENTS - 1; i++) {
    if (average[i] > 89) {
        Console.WriteLine(names[i] + " Bravo!");
    }
}
}
}

```

7. Solution

```

const int ARTISTS = 15;
const int JUDGES = 5;

static void Main(string[] args) {
    int i, j, total;

    string[] judge_names = new string[JUDGES];
    for (j = 0; j <= JUDGES - 1; j++) {
        Console.Write("Enter name for judge No " + (j + 1) + ": ");
        judge_names[j] = Console.ReadLine();
    }

    string[] artist_names = new string[ARTISTS];
    string[] song_titles = new string[ARTISTS];
    int[,] score = new int[ARTISTS, JUDGES];
    for (i = 0; i <= ARTISTS - 1; i++) {

```

```

    Console.Write("Enter name for artist No " + (i + 1) + ": ");
    artist_names[i] = Console.ReadLine();
    Console.Write("Enter song title for artist " + artist_names[i] + ": ");
    song_titles[i] = Console.ReadLine();
    for (j = 0; j <= JUDGES - 1; j++) {
        Console.Write("Enter score for artist: " + artist_names[i]);
        Console.Write(" gotten from judge " + judge_names[j] + ": ");
        score[i, j] = Int32.Parse(Console.ReadLine());
    }
}

for (i = 0; i <= ARTISTS - 1; i++) {
    total = 0;
    for (j = 0; j <= JUDGES - 1; j++) {
        total += score[i, j];
    }
    Console.WriteLine(artist_names[i] + ", " + song_titles[i] + ": " + total);
}

for (j = 0; j <= JUDGES - 1; j++) {
    total = 0;
    for (i = 0; i <= ARTISTS - 1; i++) {
        total += score[i, j];
    }
    Console.WriteLine(judge_names[j] + ": " + total / (double)ARTISTS);
}
}

```

8. Solution

```

const int PEOPLE = 30;
const int MONTHS = 12;

static void Main(string[] args) {
    int i, j, sum_heights, sum_weights;
    double average_height, average_weight;

    int[,] weights = new int[PEOPLE, MONTHS];
    int[,] heights = new int[PEOPLE, MONTHS];
    for (i = 0; i <= PEOPLE - 1; i++) {
        for (j = 0; j <= MONTHS - 1; j++) {
            weights[i, j] = Int32.Parse(Console.ReadLine());
            heights[i, j] = Int32.Parse(Console.ReadLine());
        }
    }

    for (i = 0; i <= PEOPLE - 1; i++) {
        sum_weights = 0;
        sum_heights = 0;
        for (j = 0; j <= MONTHS - 1; j++) {
            sum_weights += weights[i, j];
            sum_heights += heights[i, j];
        }
    }
}

```



```

    }
    average_weight = sum_weights / (double)MONTHS;
    average_height = sum_heights / (double)MONTHS;
    Console.WriteLine(average_weight + ", " + average_height);
    Console.WriteLine(average_weight * 702 / Math.Pow(average_height, 2));
}

for (i = 0; i <= PEOPLE - 1; i++) {
    Console.WriteLine(weights[i, 4] * 702 / Math.Pow(heights[i, 4], 2));
    Console.WriteLine(weights[i, 7] * 702 / Math.Pow(heights[i, 7], 2));
}
}

```

9. Solution

```

const double VAT = 0.19;
const int CONSUMERS = 1000;

static void Main(string[] args) {
    int consumed, i;
    double payment, total;

    int[,] meter_reading = new int[CONSUMERS, 2];
    for (i = 0; i <= CONSUMERS - 1; i++) {
        meter_reading[i, 0] = Int32.Parse(Console.ReadLine());
        meter_reading[i, 1] = Int32.Parse(Console.ReadLine());
    }

    total = 0;
    for (i = 0; i <= CONSUMERS - 1; i++) {
        consumed = meter_reading[i, 1] - meter_reading[i, 0];
        Console.WriteLine(consumed);
        payment = consumed * 0.07;
        payment += VAT * payment;
        Console.WriteLine(payment);

        total += consumed;
    }

    Console.WriteLine(total + ", " + (total * 0.07 + total * 0.07 * VAT));
}

```

10. Solution

```

const int CURRENCIES = 4;
const int DAYS = 5;

static void Main(string[] args) {
    int i, j;
    double average, total, usd;

    Console.Write("Enter an amount in US dollars: ");
    usd = Double.Parse(Console.ReadLine());
}

```

```

string[] currency = {"British Pounds Sterling", "Euros", "Canadian Dollars", "Australian Dollars"};

double[,] rate = {{1.320, 1.321, 1.332, 1.331, 1.341},
                  {1.143, 1.156, 1.138, 1.122, 1.129},
                  {0.757, 0.764, 0.760, 0.750, 0.749},
                  {0.720, 0.725, 0.729, 0.736, 0.739}
                  };

for (i = 0; i <= CURRENCIES - 1; i++) {
    total = 0;
    for (j = 0; j <= DAYS - 1; j++) {
        total += rate[i, j];
    }
    average = total / DAYS;
    Console.WriteLine(usd + " US dollars = " + (usd / average) + " " + currency[i]);
}

```

11. Solution

```

const int EMPLOYEES = 10;
const int DAYS = 5;

static void Main(string[] args) {
    int i, j;
    double gross_pay, pay_rate, total;

    string[] days = {"Monday", "Tuesday", "Wednesday", "Thursday", "Friday"};

    pay_rate = Double.Parse(Console.ReadLine());

    string[] names = new string[EMPLOYEES];
    int[,] hours_worked_per_day = new int[EMPLOYEES, DAYS];
    for (i = 0; i <= EMPLOYEES - 1; i++) {
        names[i] = Console.ReadLine();
        for (j = 0; j <= DAYS - 1; j++) {
            hours_worked_per_day[i, j] = Int32.Parse(Console.ReadLine());
        }
    }

    int[] hours_worked_per_week = new int[EMPLOYEES];
    for (i = 0; i <= EMPLOYEES - 1; i++) {
        hours_worked_per_week[i] = 0;
        for (j = 0; j <= DAYS - 1; j++) {
            hours_worked_per_week[i] += hours_worked_per_day[i, j];
        }
        if (hours_worked_per_week[i] > 40) {
            Console.WriteLine(names[i]);
        }
    }

    for (i = 0; i <= EMPLOYEES - 1; i++) {

```

```

    if (hours_worked_per_week[i] <= 40) {
        gross_pay = pay_rate * hours_worked_per_week[i];
    }
    else {
        gross_pay = pay_rate * 40 + 1.5 * pay_rate * (hours_worked_per_week[i] - 40);
    }
    Console.WriteLine(names[i] + ", " + gross_pay);
}

for (i = 0; i <= EMPLOYEES - 1; i++) {
    if (hours_worked_per_week[i] > 40) {
        for (j = 0; j <= DAYS - 1; j++) {
            if (hours_worked_per_day[i, j] > 8) {
                Console.WriteLine(names[i] + ", " + days[j] + " Overtime!");
            }
        }
    }
}

for (j = 0; j <= DAYS - 1; j++) {
    total = 0;
    for (i = 0; i <= EMPLOYEES - 1; i++) {
        if (hours_worked_per_day[i, j] <= 8) {
            gross_pay = pay_rate * hours_worked_per_day[i, j];
        }
        else {
            gross_pay = pay_rate * 8 + 1.5 * pay_rate * (hours_worked_per_day[i, j] - 8);
        }
        total += gross_pay;
    }
    Console.WriteLine(days[j] + ", " + total);
}
}

```

12. Solution

```

const int ROWS = 3;
const int COLUMNS = 4;

static void Main(string[] args) {
    int i, j, k;

    int[,] a = { {9, 9, 2, 6},
                 {4, 1, 10, 11},
                 {12, 15, 7, 3}
               };

    int[] b = new int[ROWS * COLUMNS];
    k = 0;
    for (i = 0; i <= ROWS - 1; i++) {
        for (j = 0; j <= COLUMNS - 1; j++) {
            b[k++] = a[i, j];
        }
    }
}

```

```
    }  
}  
  
for (k = 0; k <= b.length - 1; k++) {  
    Console.Write(b[k] + " ");  
}  
}
```

13. Solution

```
const int ROWS = 3;  
const int COLUMNS = 3;  
  
static void Main(string[] args) {  
    int i, j, k;  
  
    int[] a = {16, 12, 3, 5, 6, 9, 18, 19, 20};  
  
    int[,] b = new int[ROWS, COLUMNS];  
    k = 0;  
    for (i = ROWS - 1; i >= 0; i--) {  
        for (j = 0; j <= COLUMNS - 1; j++) {  
            b[i, j] = a[k++];  
        }  
    }  
  
    for (i = 0; i <= ROWS - 1; i++) {  
        for (j = 0; j <= COLUMNS - 1; j++) {  
            Console.Write(b[i, j] + "\t");  
        }  
        Console.WriteLine();  
    }  
}
```

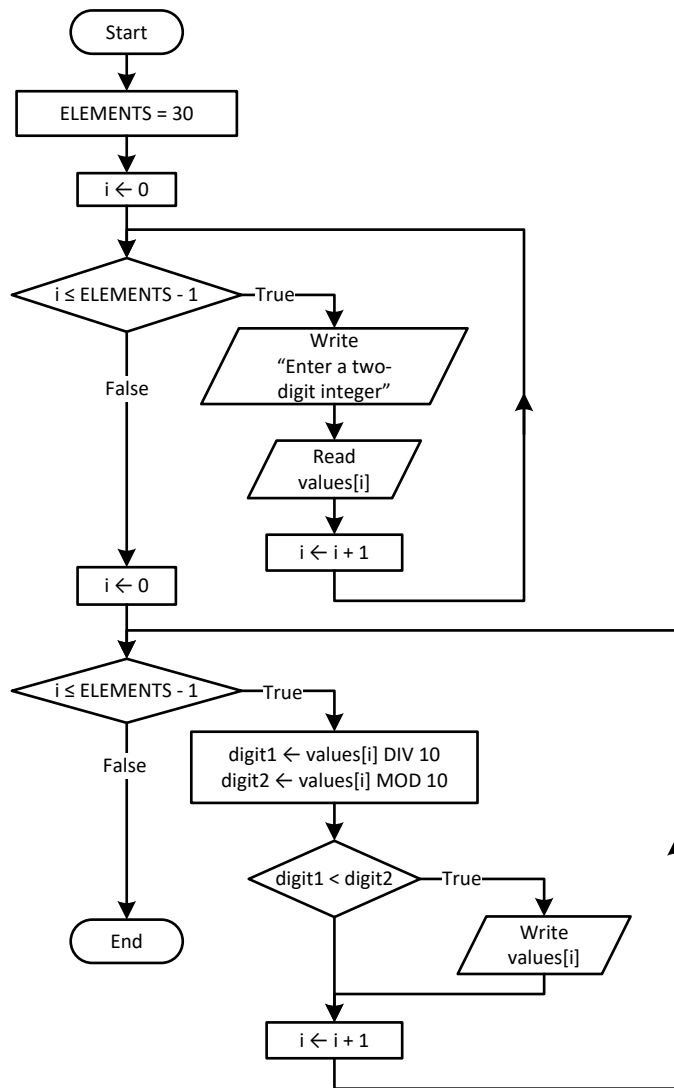
Chapter 34


34.7 Review Questions: True/False

- | | |
|-----------|-----------|
| 1. true | 21. true |
| 2. false | 22. true |
| 3. true | 23. true |
| 4. true | 24. false |
| 5. true | 25. true |
| 6. true | 26. false |
| 7. true | 27. false |
| 8. false | 28. false |
| 9. true | 29. true |
| 10. false | 30. true |
| 11. false | 31. true |
| 12. true | 32. false |
| 13. false | 33. true |
| 14. false | 34. false |
| 15. false | 35. true |
| 16. true | 36. true |
| 17. true | 37. false |
| 18. true | 38. true |
| 19. false | 39. true |
| 20. false | 40. false |

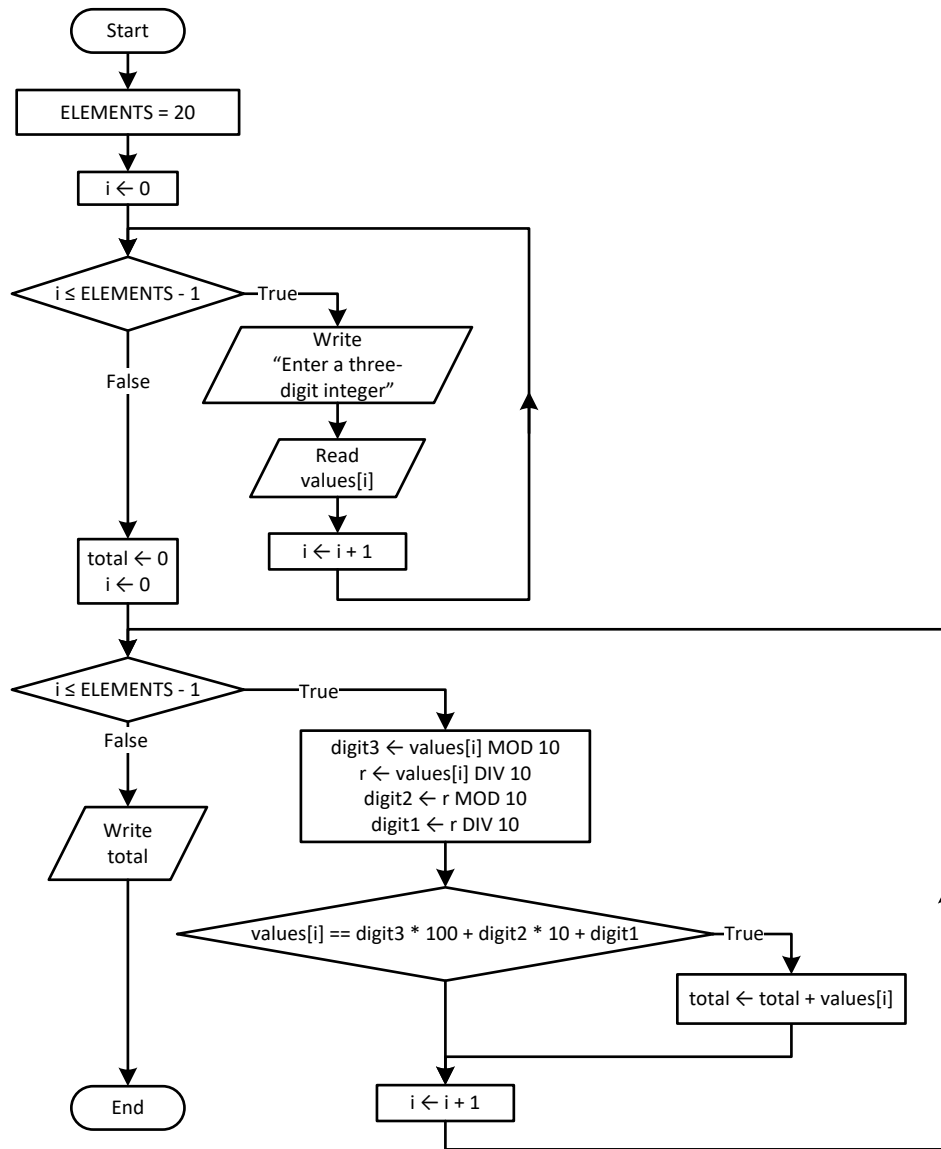
34.8 Review Exercises

1. Solution

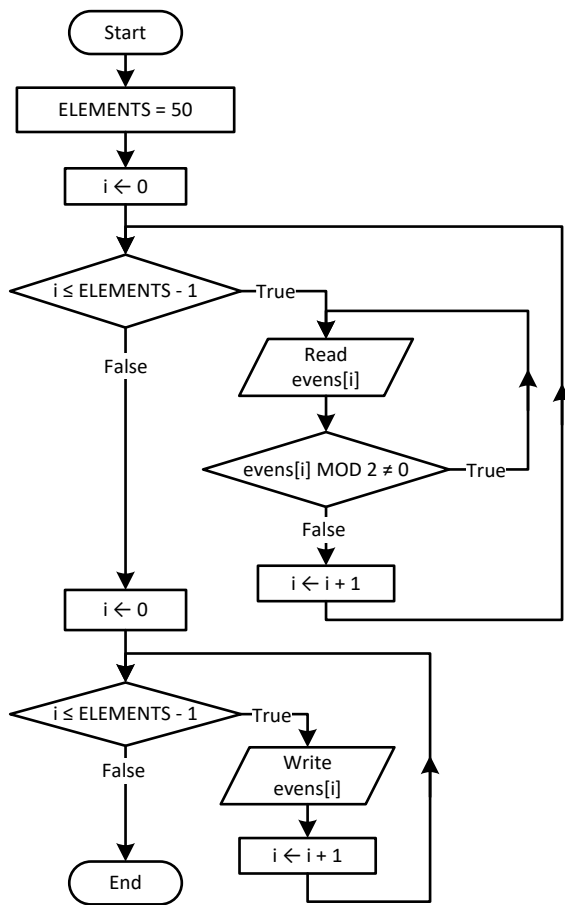


 Please note that since flowcharts are a loose method to represent an algorithm, it is not necessary to initialize an array within a flowchart; that is, there is no need to represent the statement `int[] values = new int[ELEMENTS]`.

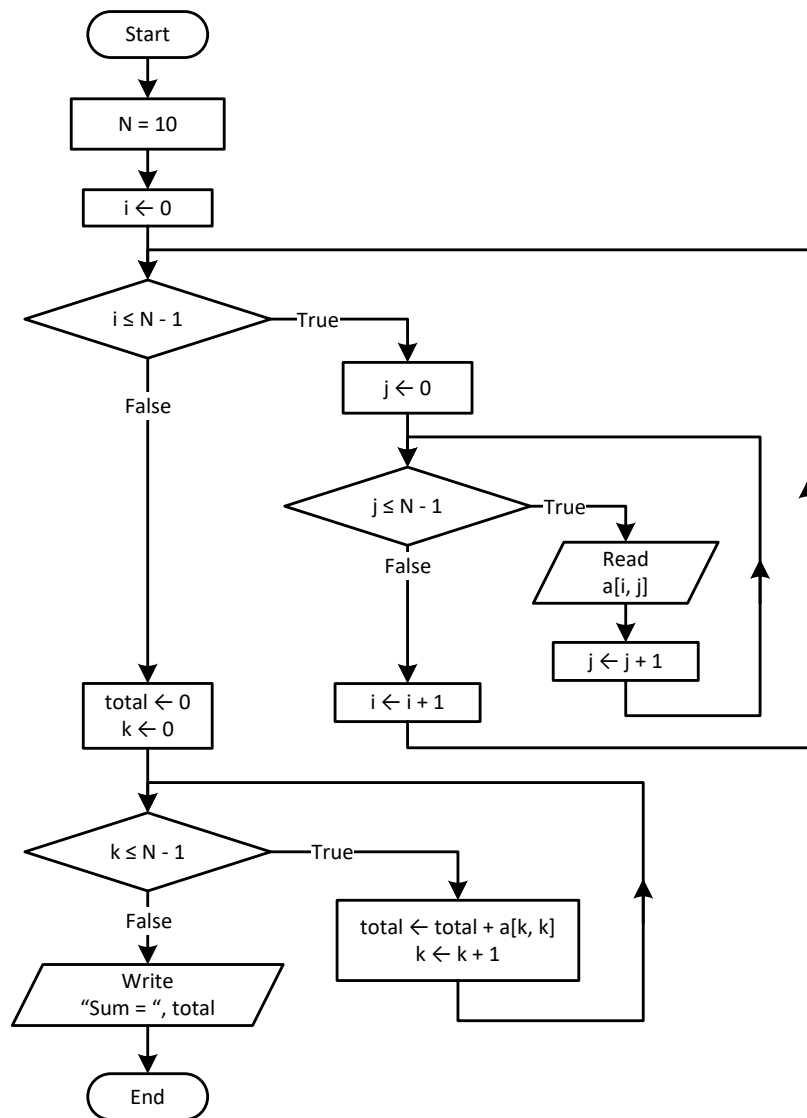
2. Solution



3. Solution



4. Solution



5. Solution

```

for (i = 0; i <= CITIES - 1; i++) {
    do {
        b[i] = Double.Parse(Console.ReadLine());
    } while (b[i] >= 0);
}

```

6. Solution

```

static void Main(string[] args) {
    int i, m, n;
    double b;
    double[] pos = new double[90];
    double[] neg = new double[90];
}

```

```
i = 1;
m = 0;
n = 0;
do {
    b = Double.Parse(Console.ReadLine());
    if (b < 0) {
        pos[m] = b;
        m++;
    }
    else {
        neg[n] = b;
        n++;
    }
    i++;
} while (i < 90);
Console.WriteLine("The End");
}
```

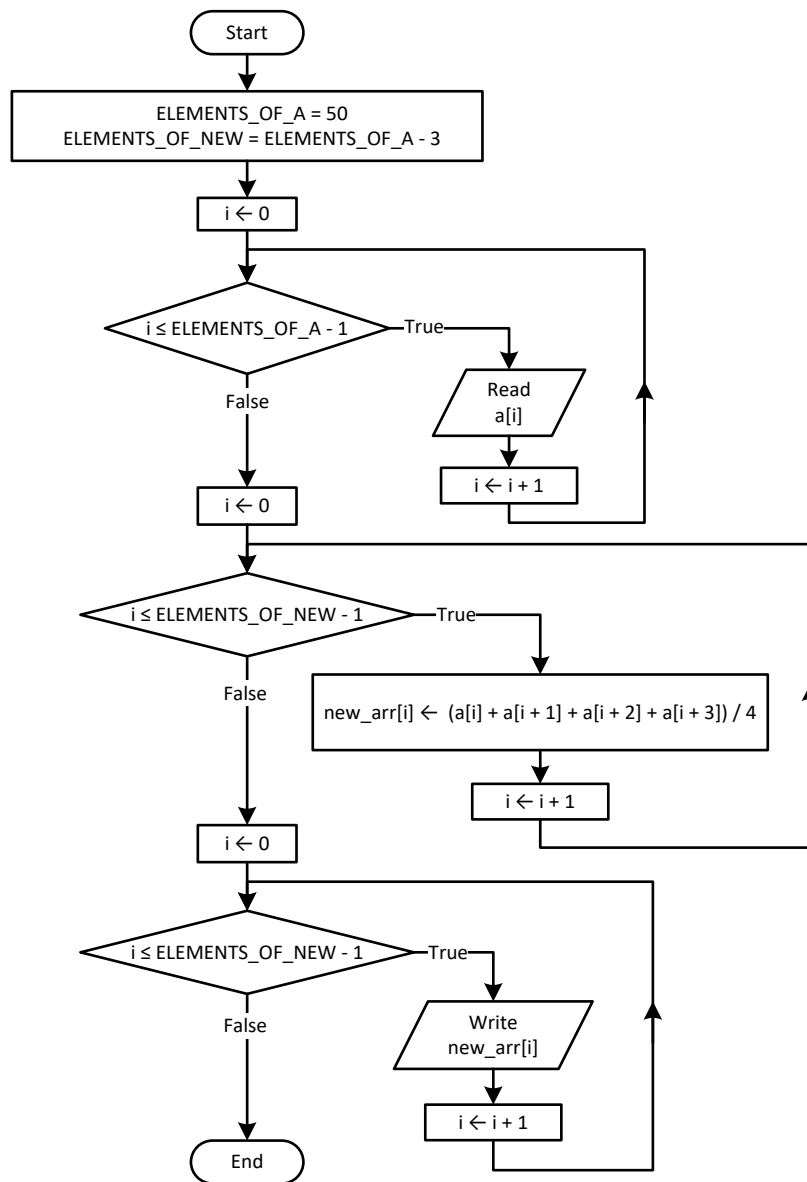
7. Solution

```
max_i = 0;
max_j = 0;
for (i = 0; i <= CITIES - 1; i++) {
    for (j = 0; j <= CITIZENS - 1; j++) {
        if (a[i, j] > a[max_i, max_j]) {
            max_i = i;
            max_j = j;
        }
    }
}
Console.WriteLine(a[max_i, max_j]);
```

8. Solution

```
for (i = 0; i <= ROWS - 1; i++) {
    for (j = 0; j <= COLUMNS - 1; j++) {
        a[i, j] = Double.Parse(Console.ReadLine());
        while (a[i, j] == 0) {
            Console.WriteLine("Error");
            a[i, j] = Double.Parse(Console.ReadLine());
        }
    }
}
```

9. Solution



```

const int ELEMENTS_OF_A = 50;
const int ELEMENTS_OF_NEW = ELEMENTS_OF_A - 3;

static void Main(string[] args) {
    int i;

    double[] a = new double[ELEMENTS_OF_A];
    for (i = 0; i <= ELEMENTS_OF_A - 1; i++) {
        a[i] = Double.Parse(Console.ReadLine());
    }

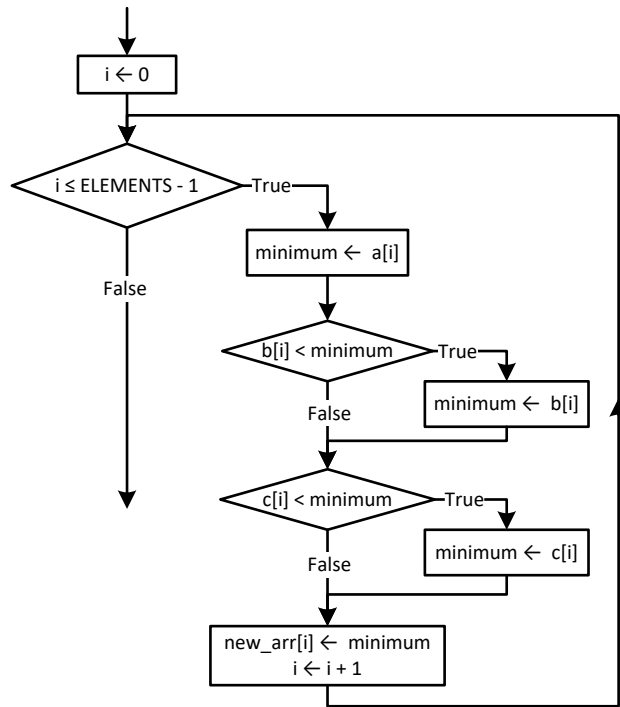
    double[] new_arr = new double[ELEMENTS_OF_NEW];
    for (i = 0; i <= ELEMENTS_OF_NEW - 1; i++) {
        new_arr[i] = (a[i] + a[i + 1] + a[i + 2] + a[i + 3]) / 4;
    }
}

```

```
    for (i = 0; i <= ELEMENTS_OF_NEW - 1; i++) {  
        Console.WriteLine(new_arr[i] + "\t");  
    }  
}
```

10. Solution

```
const int ELEMENTS = 15;  
  
static void Main(string[] args) {  
    int i;  
    double minimum  
  
    double[] a = new double[ELEMENTS];  
    for (i = 0; i <= ELEMENTS - 1; i++) {  
        a[i] = Double.Parse(Console.ReadLine());  
    }  
    double[] b = new double[ELEMENTS];  
    for (i = 0; i <= ELEMENTS - 1; i++) {  
        b[i] = Double.Parse(Console.ReadLine());  
    }  
    double[] c = new double[ELEMENTS];  
    for (i = 0; i <= ELEMENTS - 1; i++) {  
        c[i] = Double.Parse(Console.ReadLine());  
    }  
  
    double[] new_arr = new double[ELEMENTS];  
    for (i = 0; i <= ELEMENTS - 1; i++) {  
        minimum = a[i];  
        if (b[i] < minimum) {  
            minimum = b[i];  
        }  
        if (c[i] < minimum) {  
            minimum = c[i];  
        }  
        new_arr[i] = minimum;  
    }  
  
    for (i = 0; i <= ELEMENTS - 1; i++) {  
        Console.WriteLine(new_arr[i]);  
    }  
}
```



11. Solution

```

const int ELEMENTS_OF_A = 10;
const int ELEMENTS_OF_B = 5;
const int ELEMENTS_OF_C = 15;
const int ELEMENTS_OF_NEW = ELEMENTS_OF_A + ELEMENTS_OF_B + ELEMENTS_OF_C;

static void Main(string[] args) {
    int i;

    double[] a = new double[ELEMENTS_OF_A];
    for (i = 0; i <= ELEMENTS_OF_A - 1; i++) {
        a[i] = Double.Parse(Console.ReadLine());
    }
    double[] b = new double[ELEMENTS_OF_B];
    for (i = 0; i <= ELEMENTS_OF_B - 1; i++) {
        b[i] = Double.Parse(Console.ReadLine());
    }
    double[] c = new double[ELEMENTS_OF_C];
    for (i = 0; i <= ELEMENTS_OF_C - 1; i++) {
        c[i] = Double.Parse(Console.ReadLine());
    }

    double[] new_arr = new double[ELEMENTS_OF_NEW];
    for (i = 0; i <= ELEMENTS_OF_C - 1; i++) {
        new_arr[i] = c[i];
    }
    for (i = 0; i <= ELEMENTS_OF_B - 1; i++) {
        new_arr[ELEMENTS_OF_C + i] = b[i];
    }
}

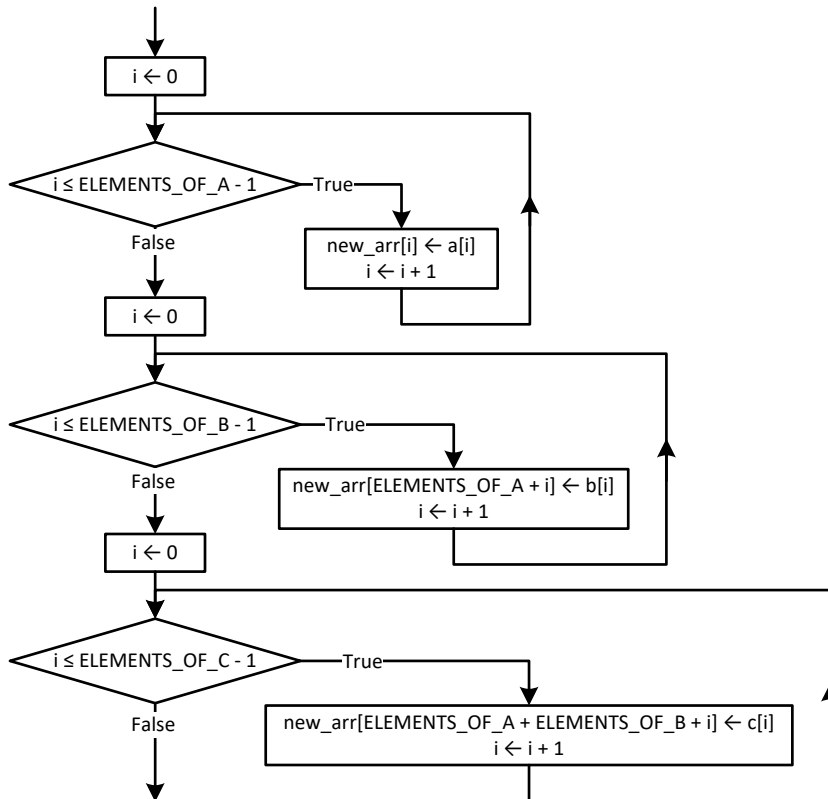
```

```

    }
    for (i = 0; i <= ELEMENTS_OF_A - 1; i++) {
        new_arr[ELEMENTS_OF_B + ELEMENTS_OF_C + i] = a[i];
    }

    //Display array new
    for (i = 0; i <= ELEMENTS_OF_NEW - 1; i++) {
        Console.Write(new_arr[i] + "\t");
    }
}

```



12. Solution

```

const int COLUMNS_OF_A = 10;
const int COLUMNS_OF_B = 15;
const int COLUMNS_OF_C = 20;
const int ROWS = 5;
const int COLUMNS = COLUMNS_OF_A + COLUMNS_OF_B + COLUMNS_OF_C;

static void Main(string[] args) {
    int i, j;

    double[,] a = new double[ROWS, COLUMNS_OF_A];
    for (i = 0; i <= ROWS - 1; i++) {
        for (j = 0; j <= COLUMNS_OF_A - 1; j++) {
            a[i, j] = Double.Parse(Console.ReadLine());
        }
    }
}

```

```

    }

    double[,] b = new double[ROWS, COLUMNS_OF_B];
    for (i = 0; i <= ROWS - 1; i++) {
        for (j = 0; j <= COLUMNS_OF_B - 1; j++) {
            b[i, j] = Double.Parse(Console.ReadLine());
        }
    }

    double[,] c = new double[ROWS, COLUMNS_OF_C];
    for (i = 0; i <= ROWS - 1; i++) {
        for (j = 0; j <= COLUMNS_OF_C - 1; j++) {
            c[i, j] = Double.Parse(Console.ReadLine());
        }
    }

    double[,] new_arr = new double[ROWS, COLUMNS];
    for (i = 0; i <= ROWS - 1; i++) {
        for (j = 0; j <= COLUMNS_OF_A - 1; j++) {
            new_arr[i, j] = a[i, j];
        }
    }
    for (i = 0; i <= ROWS - 1; i++) {
        for (j = 0; j <= COLUMNS_OF_B - 1; j++) {
            new_arr[i, COLUMNS_OF_A + j] = b[i, j];
        }
    }
    for (i = 0; i <= ROWS - 1; i++) {
        for (j = 0; j <= COLUMNS_OF_C - 1; j++) {
            new_arr[i, COLUMNS_OF_A + COLUMNS_OF_B + j] = c[i, j];
        }
    }

    for (i = 0; i <= ROWS - 1; i++) {
        for (j = 0; j <= COLUMNS - 1; j++) {
            Console.Write(new_arr[i, j] + "\t");
        }
        Console.WriteLine();
    }
}

```

13. Solution

```

const int ELEMENTS = 50;

static void Main(string[] args) {
    int i, integers_index, reals_index;

    double[] a = new double[ELEMENTS];
    for (i = 0; i <= ELEMENTS - 1; i++) {
        a[i] = Double.Parse(Console.ReadLine());
    }
}

```

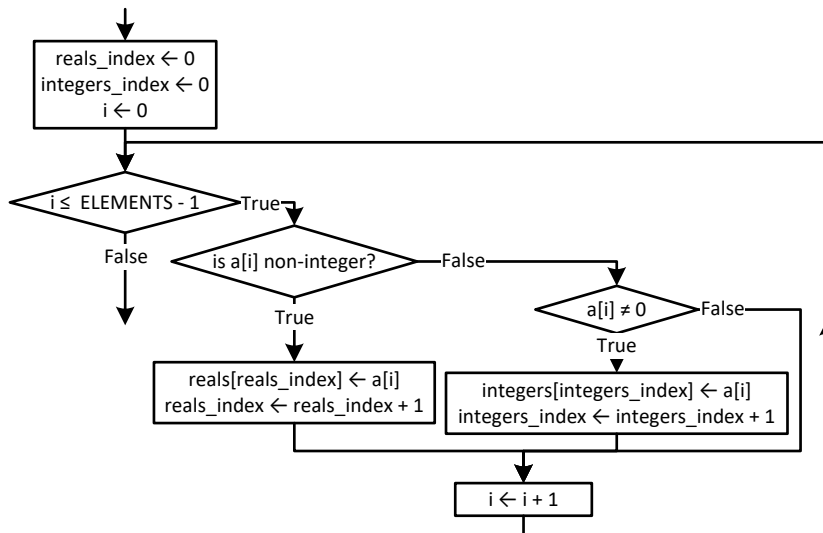
```

double[] reals = new double[ELEMENTS];
int[] integers = new int[ELEMENTS];
reals_index = 0;
integers_index = 0;
for (i = 0; i <= ELEMENTS - 1; i++) {
    if (a[i] != (int)(a[i])) {
        reals[reals_index] = a[i];
        reals_index++;
    }
    else if (a[i] != 0) {
        integers[integers_index] = (int)a[i];
        integers_index++;
    }
}

for (i = 0; i <= reals_index - 1; i++) {
    Console.Write(reals[i] + "\t");
}

Console.WriteLine();
for (i = 0; i <= integers_index - 1; i++) {
    Console.Write(integers[i] + "\t");
}
}

```



14. Solution

```

const int ELEMENTS = 50;

static void Main(string[] args) {
    int digit1, digit2, digit3, i, k, r;

    int[] a = new int[ELEMENTS];
    for (i = 0; i <= ELEMENTS - 1; i++) {
        a[i] = Int32.Parse(Console.ReadLine());
    }
}

```



```

    }

    int[] b = new int[ELEMENTS];
    k = 0;
    for (i = 0; i <= ELEMENTS - 1; i++) {
        digit3 = a[i] % 10;
        r = (int)(a[i] / 10);
        digit2 = r % 10;
        digit1 = (int)(r / 10);

        if (digit1 < digit2 && digit2 < digit3) {
            b[k] = a[i];
            k++;
        }
    }

    for (i = 0; i <= k - 1; i++) {
        Console.Write(b[i] + "\t");
    }
}

```

15. Solution

```

const int PRODUCTS = 10;
const int CITIZENS = 200;

static void Main(string[] args) {
    int count_B, i, j, maximum;

    string[] prod_names = new string[PRODUCTS];
    string[,] answers = new string[PRODUCTS, CITIZENS];
    for (i = 0; i <= PRODUCTS - 1; i++) {
        prod_names[i] = Console.ReadLine();
        for (j = 0; j <= CITIZENS - 1; j++) {
            answers[i, j] = Console.ReadLine();
            while (answers[i, j].CompareTo("A") < 0 || answers[i, j].CompareTo("D") > 0) {
                Console.WriteLine("Error! ");
                answers[i, j] = Console.ReadLine();
            }
        }
    }

    int[] count_A = new int[PRODUCTS];
    for (i = 0; i <= PRODUCTS - 1; i++) {
        count_A[i] = 0;
        for (j = 0; j <= CITIZENS - 1; j++) {
            if (answers[i, j] == "A") {
                count_A[i]++;
            }
        }
        Console.WriteLine(prod_names[i] + ", " + count_A[i]);
    }
}

```

```

    for (j = 0; j <= CITIZENS - 1; j++) {
        count_B = 0;
        for (i = 0; i <= PRODUCTS - 1; i++) {
            if (answers[i, j] == "B") {
                count_B++;
            }
        }
        Console.WriteLine(count_B);
    }

    maximum = count_A[0];
    for (i = 1; i <= PRODUCTS - 1; i++) {
        if (count_A[i] > maximum) {
            maximum = count_A[i];
        }
    }
    for (i = 0; i <= PRODUCTS - 1; i++) {
        if (count_A[i] == maximum) {
            Console.WriteLine(prod_names[i]);
        }
    }
}

```

16. Solution

```

const int US_CITIES = 20;
const int CANADIAN_CITIES = 20;

static void Main(string[] args) {
    int i, j, min_j;
    double minimum;

    string[] us_names = new string[US_CITIES];
    for (i = 0; i <= US_CITIES - 1; i++) {
        Console.WriteLine("Enter name for US city No " + (i + 1) + ": ");
        us_names[i] = Console.ReadLine();
    }

    string[] canadian_names = new string[CANADIAN_CITIES];
    for (j = 0; j <= CANADIAN_CITIES - 1; j++) {
        Console.WriteLine("Enter name for Canadian city No " + (j + 1) + ": ");
        canadian_names[j] = Console.ReadLine();
    }

    double[,] distances = new double[US_CITIES, CANADIAN_CITIES];
    for (i = 0; i <= US_CITIES - 1; i++) {
        for (j = 0; j <= CANADIAN_CITIES - 1; j++) {
            Console.WriteLine("Enter distance between " + us_names[i] + " and " + canadian_names[j] + ": ");
            distances[i, j] = Double.Parse(Console.ReadLine());
        }
    }

    for (i = 0; i <= US_CITIES - 1; i++) {

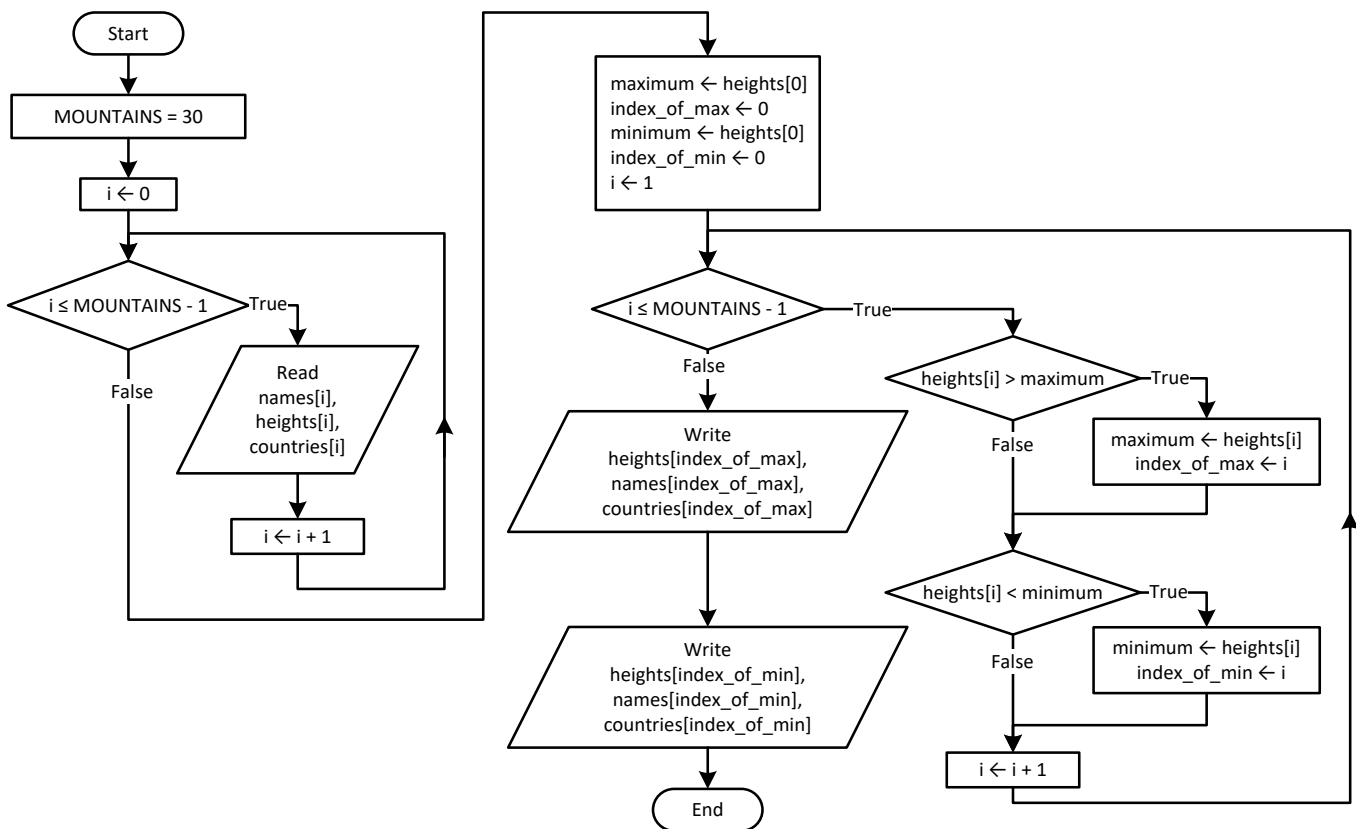
```

```

    minimum = distances[i, 0];
    min_j = 0;
    for (j = 1; j <= CANADIAN_CITIES - 1; j++) {
        if (distances[i, j] < minimum) {
            minimum = distances[i, j];
            min_j = j;
        }
    }
    Console.WriteLine("Closest Canadian city to " + us_names[i] + " is " + canadian_names[min_j]);
}
}

```

17. Solution



```

const int MOUNTAINS = 30;

static void Main(string[] args) {
    int i, index_of_max, index_of_min;
    double maximum, minimum;

    string[] names = new string[MOUNTAINS];
    double[] heights = new double[MOUNTAINS];
    string[] countries = new string[MOUNTAINS];
    for (i = 0; i <= MOUNTAINS - 1; i++) {
        names[i] = Console.ReadLine();
        heights[i] = Double.Parse(Console.ReadLine());
        countries[i] = Console.ReadLine();
    }
}

```

```

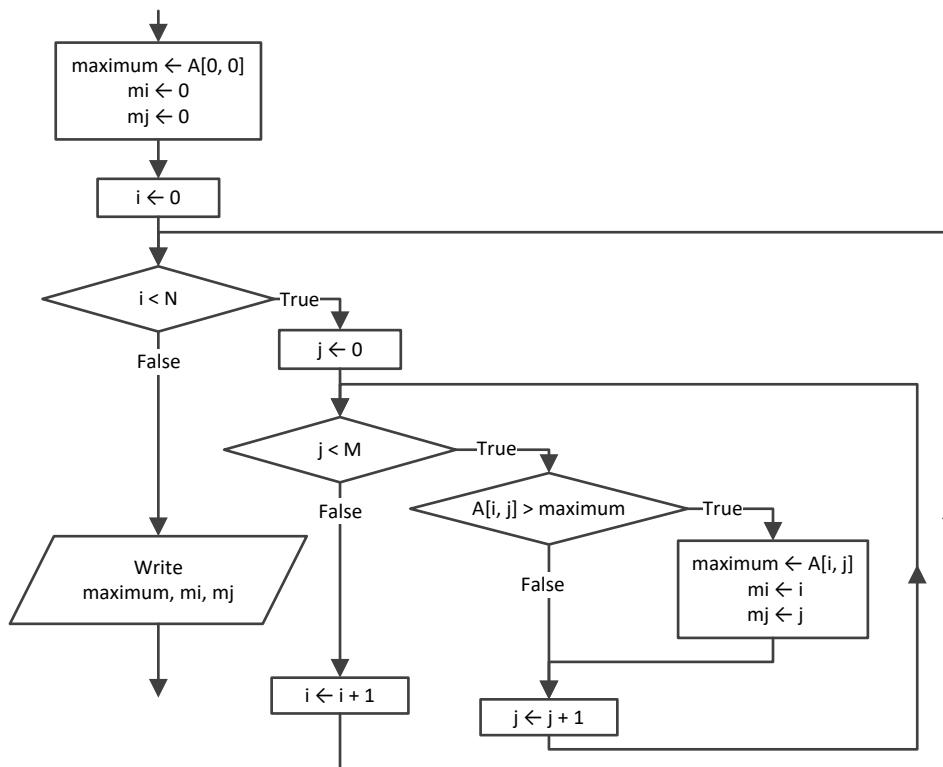
    }

    maximum = heights[0];
    index_of_max = 0;
    minimum = heights[0];
    index_of_min = 0;
    for (i = 1; i <= MOUNTAINS - 1; i++) {
        if (heights[i] > maximum) {
            maximum = heights[i];
            index_of_max = i;
        }
        if (heights[i] < minimum) {
            minimum = heights[i];
            index_of_min = i;
        }
    }

    Console.WriteLine(heights[index_of_max] + ", " + names[index_of_max] + ", " + countries[index_of_max]);
    Console.WriteLine(heights[index_of_min] + ", " + names[index_of_min] + ", " + countries[index_of_min]);
}

```

18. Solution



19. Solution

```

const int TEAMS = 26;
const int GAMES = 15;

```

```

static void Main(string[] args) {
    int i, j, m_i, maximum;

    string[] names = new string[TEAMS];
    string[,] results = new string[TEAMS, GAMES];
    for (i = 0; i <= TEAMS - 1; i++) {
        names[i] = Console.ReadLine();
        for (j = 0; j <= GAMES - 1; j++) {
            results[i, j] = Console.ReadLine();
        }
    }

    int[] points = new int[TEAMS];
    for (i = 0; i <= TEAMS - 1; i++) {
        points[i] = 0;
        for (j = 0; j <= GAMES - 1; j++) {
            if (results[i, j] == "W") {
                points[i] += 3;
            }
            else if (results[i, j] == "T") {
                points[i] += 1;
            }
        }
    }

    maximum = points[0];
    m_i = 0;
    for (i = 1; i <= TEAMS - 1; i++) {
        if (points[i] > maximum) {
            maximum = points[i];
            m_i = i;
        }
    }

    Console.WriteLine(names[m_i]);
}

```

20. Solution

```

const int OBJECTS = 10;
const int FALLS = 20;

static void Main(string[] args) {
    int i, j;
    double maxi, mini;

    double[,] heights = new double[OBJECTS, FALLS];
    double[,] times = new double[OBJECTS, FALLS];
    for (i = 0; i <= OBJECTS - 1; i++) {
        for (j = 0; j <= FALLS - 1; j++) {
            heights[i, j] = Double.Parse(Console.ReadLine());
            times[i, j] = Double.Parse(Console.ReadLine());
        }
    }
}

```

```

    }

    double[,] g = new double[OBJECTS, FALLS];
    for (i = 0; i <= OBJECTS - 1; i++) {
        for (j = 0; j <= FALLS - 1; j++) {
            g[i, j] = 2 * heights[i, j] / Math.Pow(times[i, j], 2);
        }
    }

    double[] minimum = new double[OBJECTS];
    double[] maximum = new double[OBJECTS];
    for (i = 0; i <= OBJECTS - 1; i++) {
        minimum[i] = g[i, 0];
        maximum[i] = g[i, 0];
        for (j = 1; j <= FALLS - 1; j++) {
            if (g[i, j] < minimum[i]) {
                minimum[i] = g[i, j];
            }
            if (g[i, j] > maximum[i]) {
                maximum[i] = g[i, j];
            }
        }
    }

    for (i = 0; i <= OBJECTS - 1; i++) {
        Console.WriteLine(minimum[i] + ", " + maximum[i]);
    }

    maxi = maximum[0];
    mini = minimum[0];
    for (i = 1; i <= OBJECTS - 1; i++) {
        if (maximum[i] > maxi) {
            maxi = maximum[i];
        }
        if (minimum[i] < mini) {
            mini = minimum[i];
        }
    }

    Console.WriteLine(mini + ", " + maxi);
}

```

21. Solution

```

const int STATIONS = 10;
const int DAYS = 365;

static void Main(string[] args) {
    int i, j, m_i;
    double minimum;

    string[] names = new string[STATIONS];
    double[,] co2 = new double[STATIONS, DAYS];

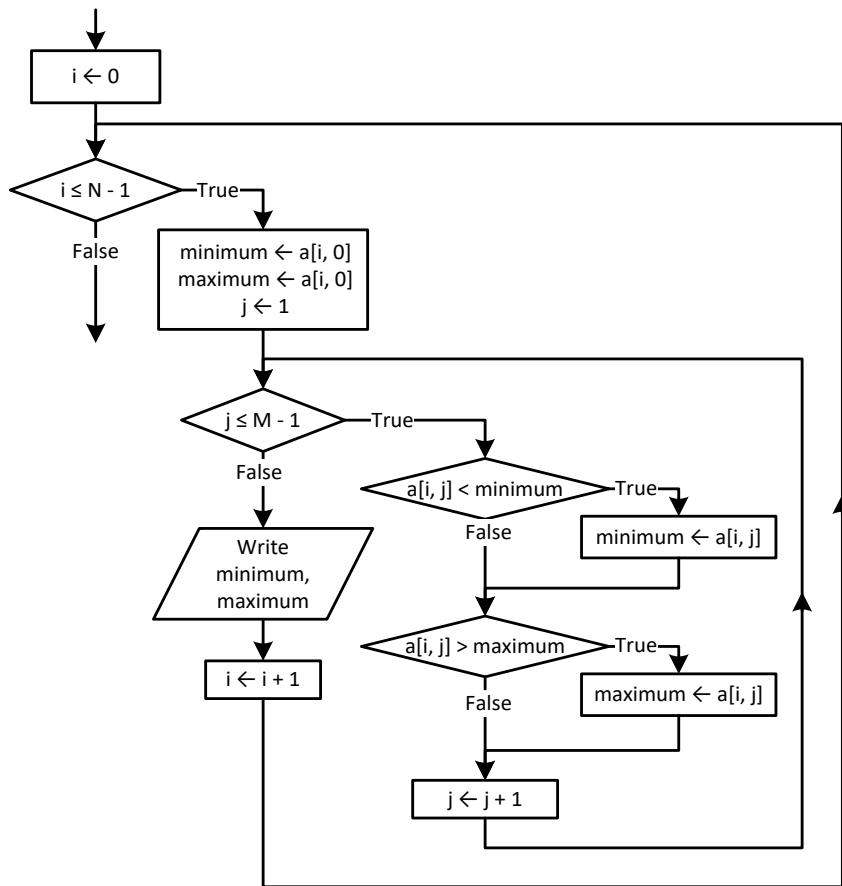
```

```
for (i = 0; i <= STATIONS - 1; i++) {
    names[i] = Console.ReadLine();
    for (j = 0; j <= DAYS - 1; j++) {
        co2[i, j] = Double.Parse(Console.ReadLine());
    }
}

double[] average = new double[STATIONS];
for (i = 0; i <= STATIONS - 1; i++) {
    average[i] = 0;
    for (j = 0; j <= DAYS - 1; j++) {
        average[i] += co2[i, j];
    }
    average[i] /= DAYS;
}

minimum = average[0];
m_i = 0;
for (i = 1; i <= STATIONS - 1; i++) {
    if (average[i] < minimum) {
        minimum = average[i];
        m_i = i;
    }
}
Console.WriteLine(names[m_i]);
}
```

22. Solution



23. Solution

First approach

```

const int ROWS = 20;
const int COLUMNS = 30;

static void Main(string[] args) {
    int i, j;

    double[,] b = new double[ROWS, COLUMNS];
    for (i = 0; i <= ROWS - 1; i++) {
        for (j = 0; j <= COLUMNS - 1; j++) {
            b[i, j] = Double.Parse(Console.ReadLine());
        }
    }

    double[] minimum = new double[COLUMNS];
    double[] maximum = new double[COLUMNS];
    for (j = 0; j <= COLUMNS - 1; j++) {
        minimum[j] = b[0, j];
        maximum[j] = b[0, j];
        for (i = 1; i <= ROWS - 1; i++) {

```



```

        if (b[i, j] < minimum[j]) {
            minimum[j] = b[i, j];
        }
        if (b[i, j] > maximum[j]) {
            maximum[j] = b[i, j];
        }
    }
}

for (j = 0; j <= COLUMNS - 1; j++) {
    Console.WriteLine(minimum[j] + " " + maximum[j]);
}
}

```

Second approach

```

const int ROWS = 20;
const int COLUMNS = 30;

static void Main(string[] args) {
    int i, j;
    double minimum, maximum;

    double[,] b = new double[ROWS, COLUMNS];
    for (i = 0; i <= ROWS - 1; i++) {
        for (j = 0; j <= COLUMNS - 1; j++) {
            b[i, j] = Double.Parse(Console.ReadLine());
        }
    }

    for (j = 0; j <= COLUMNS - 1; j++) {
        minimum = b[0, j];
        maximum = b[0, j];
        for (i = 1; i <= ROWS - 1; i++) {
            if (b[i, j] < minimum) {
                minimum = b[i, j];
            }
            if (b[i, j] > maximum) {
                maximum = b[i, j];
            }
        }
        Console.WriteLine(minimum + " " + maximum);
    }
}

```

24. Solution

```

const int TEAMS = 20;
const int GAMES = 10;

static void Main(string[] args) {
    int i, j, m, n, temp;
    bool swaps;
    string temp_str;
}

```

```

string[] names = new string[TEAMS];
string[,] results = new string[TEAMS, GAMES];
for (i = 0; i <= TEAMS - 1; i++) {
    Console.Write("Enter team name: ");
    names[i] = Console.ReadLine();
    for (j = 0; j <= GAMES - 1; j++) {
        Console.WriteLine("Enter result for team " + names[i] + " for game No " + (j + 1) + ": ");
        results[i, j] = Console.ReadLine();
        while (results[i, j] != "W" && results[i, j] != "L" && results[i, j] != "T") {
            Console.Write("Error! Enter only value W, L, or T: ");
            results[i, j] = Console.ReadLine();
        }
    }
}

int[] points = new int[TEAMS];
for (i = 0; i <= TEAMS - 1; i++) {
    points[i] = 0;
    for (j = 0; j <= GAMES - 1; j++) {
        if (results[i, j] == "W") {
            points[i] += 3;
        }
        else if (results[i, j] == "T") {
            points[i] += 1;
        }
    }
}

for (m = 1; m <= TEAMS - 1; m++) {
    swaps = false;
    for (n = TEAMS - 1; n >= m; n--) {
        if (points[n] > points[n - 1]) {
            temp = points[n];
            points[n] = points[n - 1];
            points[n - 1] = temp;

            temp_str = names[n];
            names[n] = names[n - 1];
            names[n - 1] = temp_str;

            swaps = true;
        }
    }
    if (!swaps) break;
}

Console.WriteLine("Gold: " + names[0]);
Console.WriteLine("Silver: " + names[1]);
Console.WriteLine("Bronze: " + names[2]);
}

```

25. Solution

```

const int PEOPLE = 50;

static void Main(string[] args) {
    int i, m, n;
    double temp;
    string temp_str;

    string[] names = new string[PEOPLE];
    double[] heights = new double[PEOPLE];
    for (i = 0; i <= PEOPLE - 1; i++) {
        Console.Write("Enter name for person No. " + (i + 1) + ": ");
        names[i] = Console.ReadLine();
        Console.Write("Enter height for person No. " + (i + 1) + ": ");
        heights[i] = Double.Parse(Console.ReadLine());
    }

    for (m = 1; m <= PEOPLE - 1; m++) {
        for (n = PEOPLE - 1; n >= m; n--) {
            if (heights[n] > heights[n - 1]) {
                temp = heights[n];
                heights[n] = heights[n - 1];
                heights[n - 1] = temp;

                temp_str = names[n];
                names[n] = names[n - 1];
                names[n - 1] = temp_str;
            }
            else if (heights[n] == heights[n - 1]) {
                if (names[n].CompareTo(names[n - 1]) < 0) {
                    temp_str = names[n];
                    names[n] = names[n - 1];
                    names[n - 1] = temp_str;
                }
            }
        }
    }

    for (i = 0; i <= PEOPLE - 1; i++) {
        Console.WriteLine(heights[i] + "\t" + names[i]);
    }
}

```

26. Solution

```

const int ARTISTS = 12;
const int JUDGES = 10;

static void Main(string[] args) {
    int i, j, m, maximum, minimum, n, temp;
    string temp_str;

```

```

string[] artist_names = new string[ARTISTS];
int[,] score = new int[ARTISTS, JUDGES];
for (i = 0; i <= ARTISTS - 1; i++) {
    Console.WriteLine("Enter name for artist No " + (i + 1) + ": ");
    artist_names[i] = Console.ReadLine();
    for (j = 0; j <= JUDGES - 1; j++) {
        Console.Write("Enter score for artist: " + artist_names[i]);
        Console.WriteLine(" gotten from judge No " + (j + 1) + ": ");
        score[i, j] = Int32.Parse(Console.ReadLine());
    }
}

int[] total = new int[ARTISTS];
for (i = 0; i <= ARTISTS - 1; i++) {
    total[i] = 0;
    for (j = 1; j <= JUDGES - 1; j++) {
        total[i] += score[i, j];
    }
}

for (i = 0; i <= ARTISTS - 1; i++) {
    minimum = score[i, 0];
    maximum = score[i, 0];
    for (j = 1; j <= JUDGES - 1; j++) {
        if (score[i, j] < minimum) {
            minimum = score[i, j];
        }
        if (score[i, j] > maximum) {
            maximum = score[i, j];
        }
    }
    total[i] = total[i] - minimum - maximum;
    Console.WriteLine(total[i]);
}

for (m = 1; m <= ARTISTS - 1; m++) {
    for (n = ARTISTS - 1; n >= m; n--) {
        if (total[n] > total[n - 1]) {
            temp = total[n];
            total[n] = total[n - 1];
            total[n - 1] = temp;

            temp_str = artist_names[n];
            artist_names[n] = artist_names[n - 1];
            artist_names[n - 1] = temp_str;
        }
        else if (total[n] == total[n - 1]) {
            if (artist_names[n].CompareTo(artist_names[n - 1]) < 0) {
                temp_str = artist_names[n];
                artist_names[n] = artist_names[n - 1];
                artist_names[n - 1] = temp_str;
            }
        }
    }
}

```

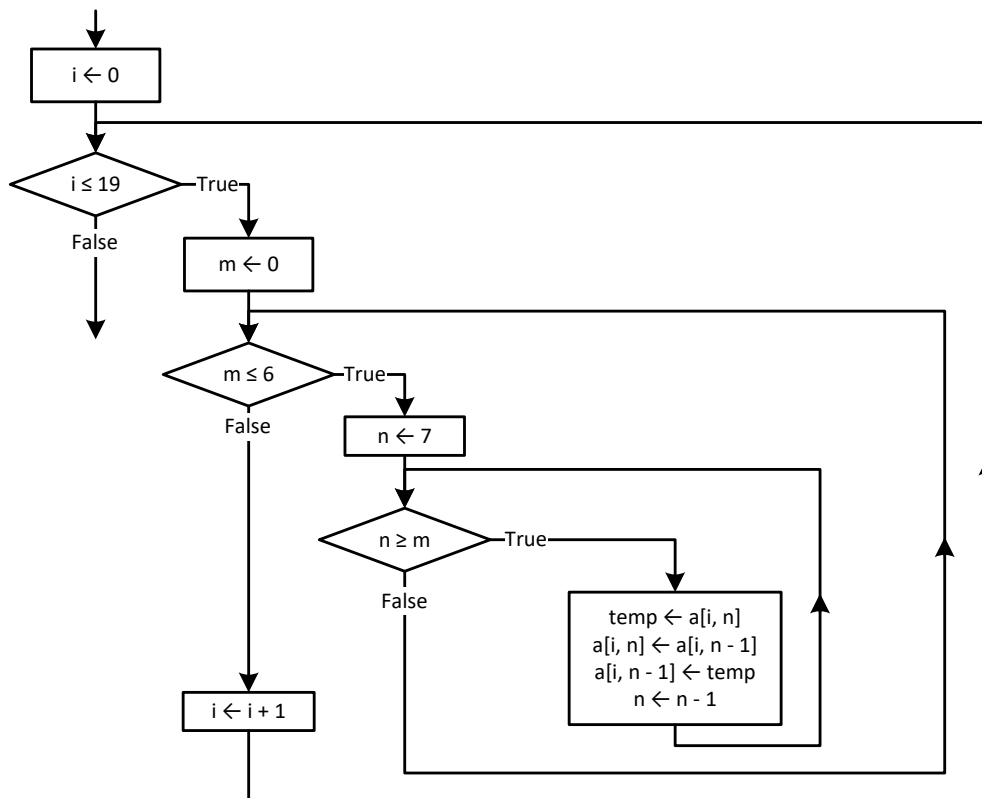
```

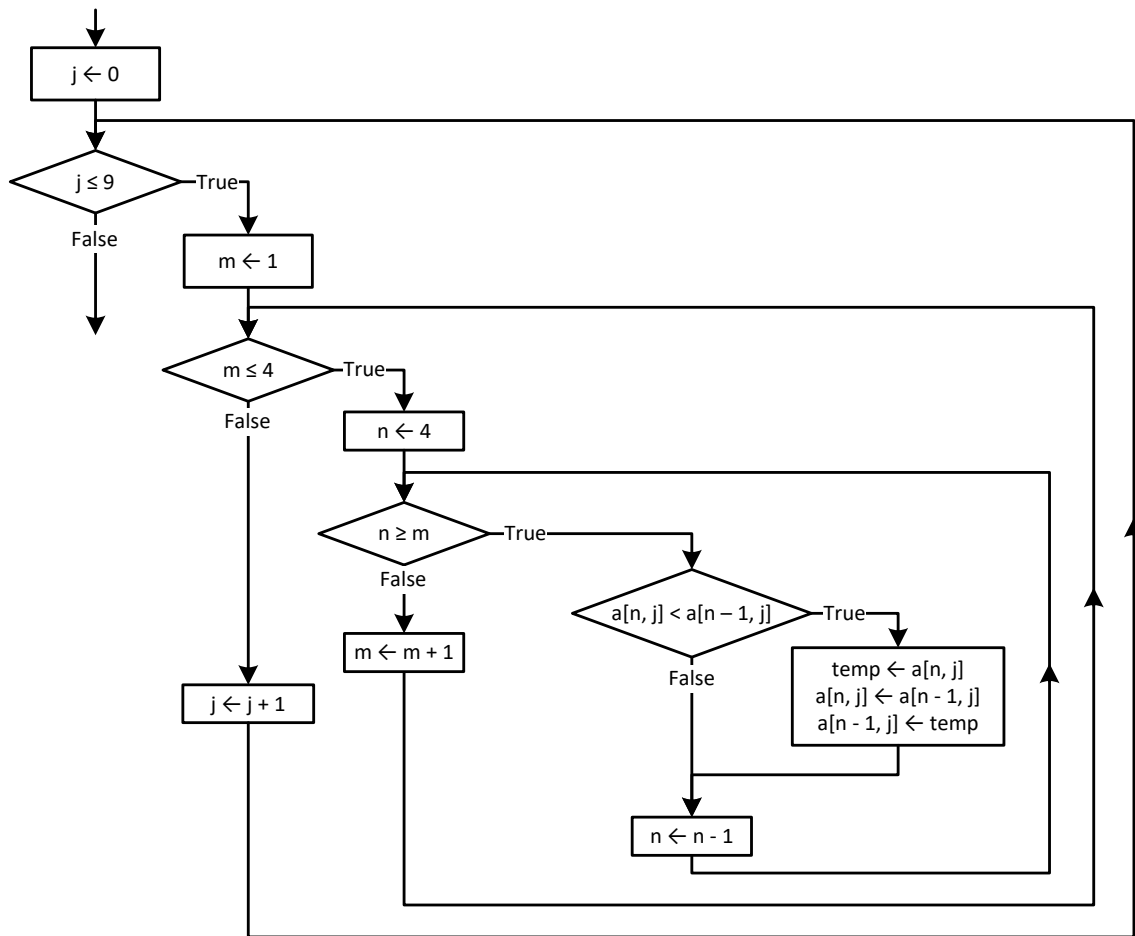
    }
  }
}

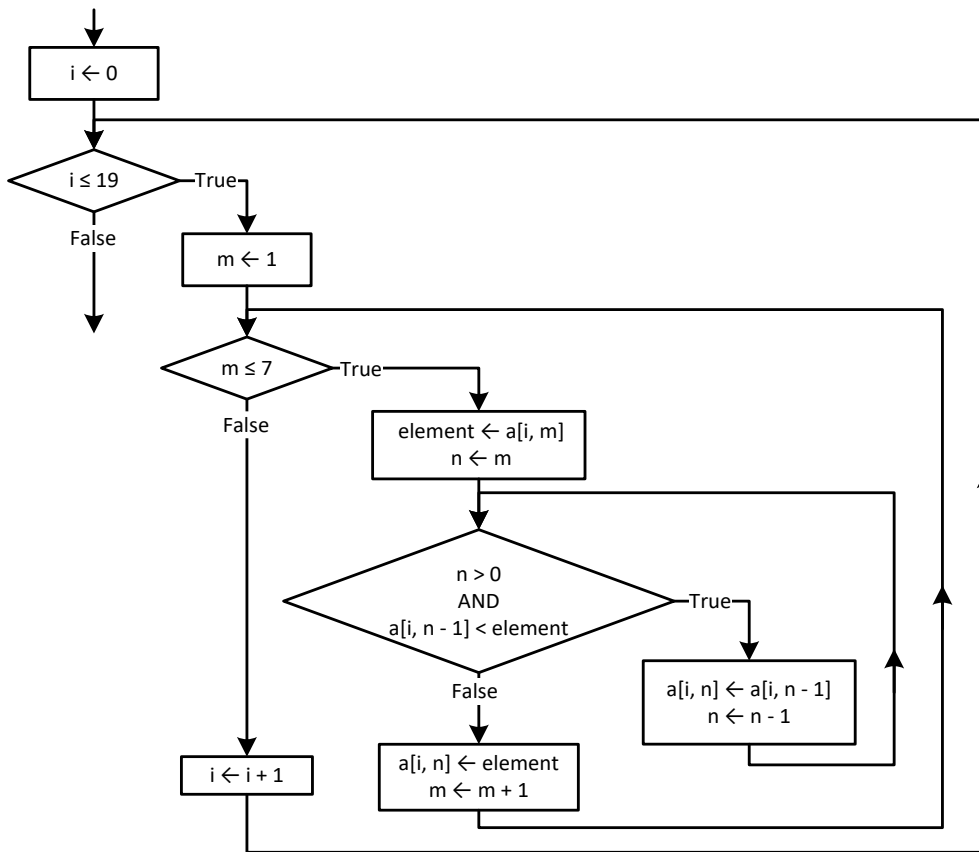
for (i = 0; i <= ARTISTS - 1; i++) {
  Console.WriteLine(artist_names[i] + ", " + total[i]);
}
}

```

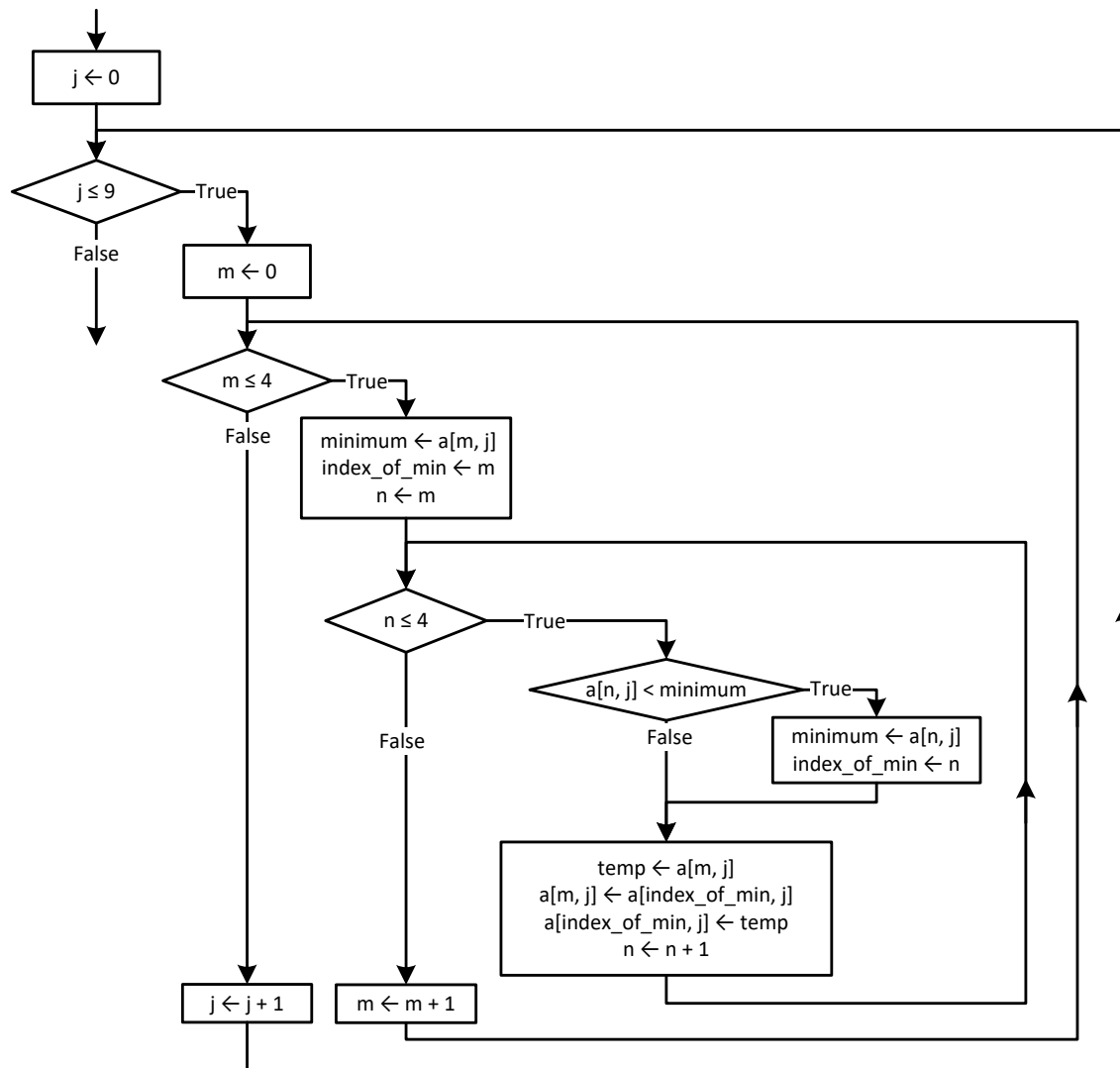
27. Solution



28. Solution

29. Solution

30. Solution



31. Solution

```

const int PEOPLE = 10;
const int PUZZLES = 8;

static void Main(string[] args) {
    int i, index_of_min, j, m, n;
    double minimum, temp;
    string temp_str;

    string[] names = new string[PEOPLE];
    double[,] times = new double[PEOPLE, PUZZLES];
    for (i = 0; i <= PEOPLE - 1; i++) {
        names[i] = Console.ReadLine();
        for (j = 0; j <= PUZZLES - 1; j++) {
            times[i, j] = Double.Parse(Console.ReadLine());
        }
    }
}

```



```

    }
}

for (i = 0; i <= PEOPLE - 1; i++) {
    for (m = 0; m <= PUZZLES - 1; m++) {
        minimum = times[i, m];
        index_of_min = m;
        for (n = m; n <= PUZZLES - 1; n++) {
            if (times[i, n] < minimum) {
                minimum = times[i, n];
                index_of_min = n;
            }
        }
        temp = times[i, m];
        times[i, m] = times[i, index_of_min];
        times[i, index_of_min] = temp;
    }
}

for (i = 0; i <= PEOPLE - 1; i++) {
    Console.WriteLine(names[i]);
    for (j = 0; j <= 2; j++) {
        Console.WriteLine(times[i, j]);
    }
}

double[] average = new double[PEOPLE];
for (i = 0; i <= PEOPLE - 1; i++) {
    average[i] = 0;
    for (j = 0; j <= PUZZLES - 1; j++) {
        average[i] += times[i, j];
    }
    average[i] /= PUZZLES;
}

for (m = 0; m <= PEOPLE - 1; m++) {
    minimum = average[m];
    index_of_min = m;
    for (n = m; n <= PEOPLE - 1; n++) {
        if (average[n] < minimum) {
            minimum = average[n];
            index_of_min = n;
        }
    }
    temp = average[m];
    average[m] = average[index_of_min];
    average[index_of_min] = temp;

    temp_str = names[m];
    names[m] = names[index_of_min];
    names[index_of_min] = temp_str;
}

```

```

    Console.WriteLine(names[0] + ", " + names[1] + ", " + names[2]);
}

```

32. Solution

```

const int AREAS = 5;
const int HOURS = 48;

static void Main(string[] args) {
    int i, j, m, m_i, m_j, n;
    double maximum, element_1;
    string element_2;

    string[] names = new string[AREAS];
    double[,] CO2 = new double[AREAS, HOURS];
    for (i = 0; i <= AREAS - 1; i++) {
        names[i] = Console.ReadLine();
        for (j = 0; j <= HOURS - 1; j++) {
            CO2[i, j] = Double.Parse(Console.ReadLine());
        }
    }

    double[] average_per_hour = new double[AREAS];
    for (i = 0; i <= AREAS - 1; i++) {
        average_per_hour[i] = 0;
        for (j = 0; j <= HOURS - 1; j++) {
            average_per_hour[i] += CO2[i, j];
        }
        average_per_hour[i] /= HOURS;
    }

    for (i = 0; i <= AREAS - 1; i++) {
        Console.WriteLine(names[i] + ", " + average_per_hour[i]);
    }

    double[] average_per_city = new double[HOURS];
    for (j = 0; j <= HOURS - 1; j++) {
        average_per_city[j] = 0;
        for (i = 0; i <= AREAS - 1; i++) {
            average_per_city[j] += CO2[i, j];
        }
        average_per_city[j] /= AREAS;
    }

    for (j = 0; j <= HOURS - 1; j++) {
        Console.WriteLine(average_per_city[j]);
    }

    maximum = average_per_city[0];
    m_j = 0;
    for (j = 1; j <= HOURS - 1; j++) {
        if (average_per_city[j] > maximum) {

```

```
        maximum = average_per_city[j];
        m_j = j;
    }
}
Console.WriteLine(m_j);

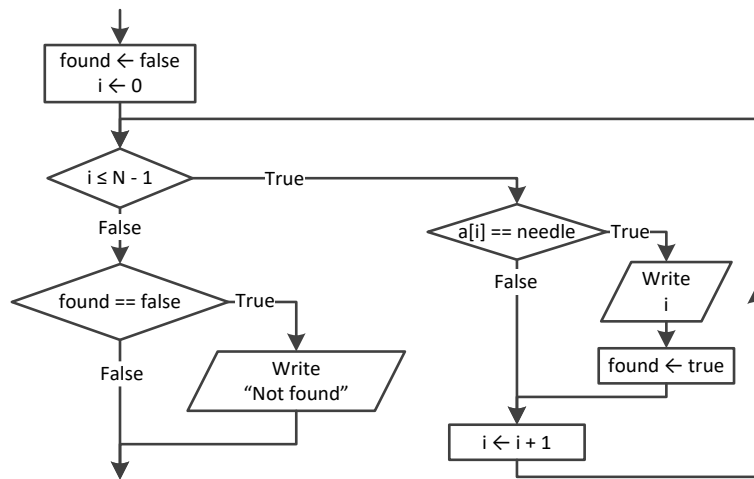
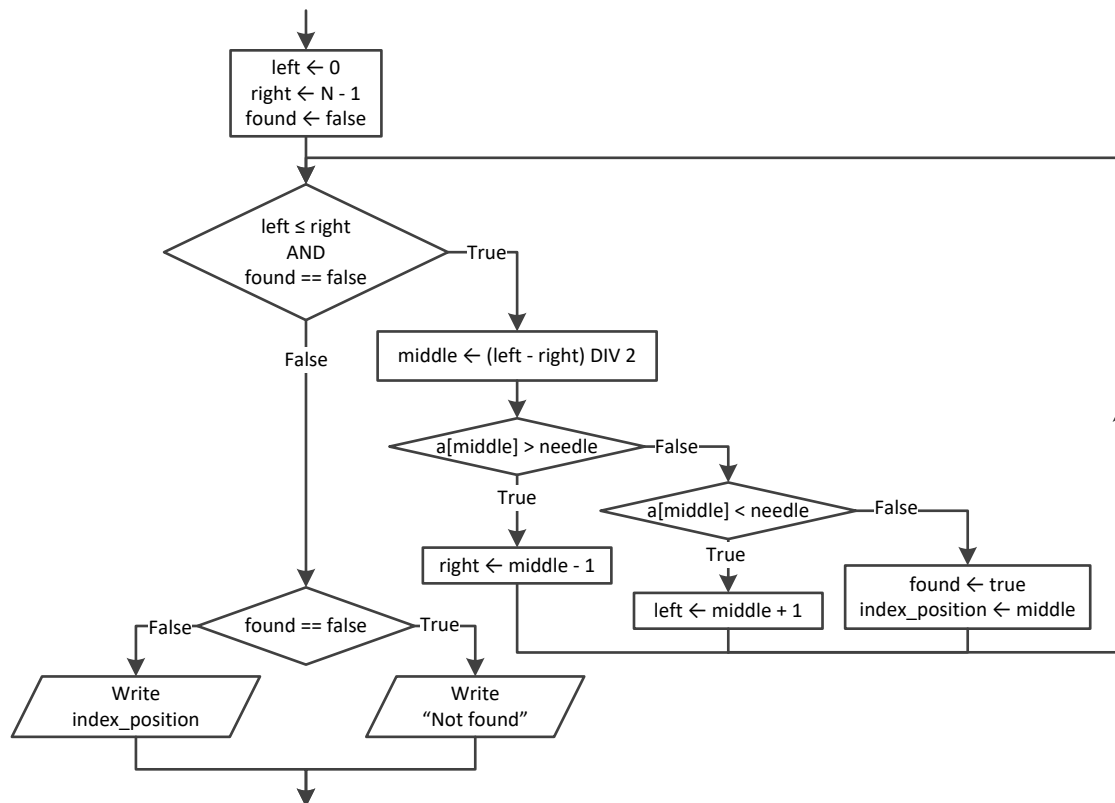
maximum = CO2[0, 0];
m_i = 0;
m_j = 0;
for (i = 0; i <= AREAS - 1; i++) {
    for (j = 0; j <= HOURS - 1; j++) {
        if (CO2[i, j] > maximum) {
            maximum = CO2[i, j];
            m_i = i;
            m_j = j;
        }
    }
}
Console.WriteLine(m_j + ", " + names[m_i]);

for (m = 1; m <= AREAS - 1; m++) {
    element_1 = average_per_hour[m];
    element_2 = names[m];

    n = m;
    while (n > 0 && average_per_hour[n - 1] < element_1) {
        average_per_hour[n] = average_per_hour[n - 1];
        names[n] = names[n - 1];
        n--;
    }

    average_per_hour[n] = element_1;
    names[n] = element_2;
}

Console.WriteLine(names[0] + ", " + names[1] + ", " + names[2]);
}
```

33. Solution**34. Solution****35. Solution**

```

const int TEAMS = 10;
const int GAMES = 16;

static void Main(string[] args) {

```

```

int i, j, total;
string needle;

string[] names = new string[TEAMS];
int[,] goals_scored = new int[TEAMS, GAMES];
int[,] goals_let_in = new int[TEAMS, GAMES];
for (i = 0; i <= TEAMS - 1; i++) {
    Console.Write("Enter team name: ");
    names[i] = Console.ReadLine();
    for (j = 0; j <= GAMES - 1; j++) {
        Console.Write("Enter goals scored: ");
        goals_scored[i, j] = Int32.Parse(Console.ReadLine());
        while (goals_scored[i, j] < 0) {
            Console.Write("Error! Enter goals scored: ");
            goals_scored[i, j] = Int32.Parse(Console.ReadLine());
        }

        Console.Write("Enter goals let in: ");
        goals_let_in[i, j] = Int32.Parse(Console.ReadLine());
        while (goals_let_in[i, j] < 0) {
            Console.Write("Error! Enter goals let in: ");
            goals_let_in[i, j] = Int32.Parse(Console.ReadLine());
        }
    }
}

Console.Write("Enter a team to search: ");
needle = Console.ReadLine();

i = 0;
while (i < TEAMS - 1 && names[i] != needle) {
    i++;
}

if (names[i] != needle) {
    Console.WriteLine("This team does not exist");
}
else {
    total = 0;
    for (j = 0; j <= GAMES - 1; j++) {
        if (goals_scored[i, j] > goals_let_in[i, j]) {
            total += 3;
        }
        else if (goals_scored[i, j] == goals_let_in[i, j]) {
            total += 1;
        }
    }
    Console.WriteLine(total);
}
}

```

36. Solution

```
const int CLASS1 = 20;
const int CLASS2 = 25;

static void Main(string[] args) {
    int i, left, m, middle, n, right;
    string temp, needle;
    bool found;

    Console.WriteLine("Class 1");
    string[] names1 = new string[CLASS1];
    for (i = 0; i <= CLASS1 - 1; i++) {
        Console.Write("Enter name: ");
        names1[i] = Console.ReadLine();
    }
    Console.WriteLine("Class 2");
    string[] names2 = new string[CLASS2];
    for (i = 0; i <= CLASS2 - 1; i++) {
        Console.Write("Enter name: ");
        names2[i] = Console.ReadLine();
    }

    //Bubble sort
    for (m = 1; m <= CLASS1 - 1; m++) {
        for (n = CLASS1 - 1; n >= m; n--) {
            if (names1[n].CompareTo(names1[n - 1]) < 0) {
                temp = names1[n];
                names1[n] = names1[n - 1];
                names1[n] = temp;
            }
        }
    }
    for (m = 1; m <= CLASS2 - 1; m++) {
        for (n = CLASS2 - 1; n >= m; n--) {
            if (names2[n].CompareTo(names2[n - 1]) < 0) {
                temp = names2[n];
                names2[n] = names2[n - 1];
                names2[n] = temp;
            }
        }
    }

    Console.WriteLine("\nClass 1");
    for (i = 0; i <= CLASS1 - 1; i++) {
        Console.WriteLine(names1[i]);
    }
    Console.WriteLine("\nClass 2");
    for (i = 0; i <= CLASS2 - 1; i++) {
        Console.WriteLine(names2[i]);
    }
}
```

```

Console.Write("Enter a name to search: ");
needle = Console.ReadLine();

left = 0;
right = CLASS1 - 1;
found = false;
while (left <= right && !found) {
    middle = (int)((left + right) / 2);

    if (names1[middle].CompareTo(needle) > 0) {
        right = middle - 1;
    }
    else if (names1[middle].CompareTo(needle) < 0) {
        left = middle + 1;
    }
    else {
        found = true;
    }
}

if (found) {
    Console.WriteLine("Student found in Class No 1");
}
else {
    left = 0;
    right = CLASS2 - 1;
    while (left <= right && !found) {
        middle = (int)((left + right) / 2);

        if (names2[middle].CompareTo(needle) > 0) {
            right = middle - 1;
        }
        else if (names2[middle].CompareTo(needle) < 0) {
            left = middle + 1;
        }
        else {
            found = true;
        }
    }

    if (found) {
        Console.WriteLine("Student found in Class No 2");
    }
    else {
        Console.WriteLine("Student not found in either class");
    }
}
}

```

37. Solution

```

Console.Write("Enter username: ");

```

```

usr = Console.ReadLine().ToUpper();
Console.Write("Enter password: ");
pwd = Console.ReadLine().ToUpper();

i = 0;
while (i < 99 && usernames[i].ToUpper() != usr) {
    i++;
}

if (usernames[i].ToUpper() == usr && passwords[i].ToUpper() == pwd) {
    Console.WriteLine("Login OK!");
}
else {
    Console.WriteLine("Login Failed!");
}

```

38. Solution

```

Console.Write("Enter a value to search: ");
value_str = Console.ReadLine();

found = false;

for (i = 0; i <= 999; i++) {
    if (names[i] == value_str) {
        Console.WriteLine(SSNs[i]);
        found = true;
    }
}

if (!found) {
    value = Int32.Parse(value_str);
    i = 0;
    while (i < 999 && SSNs[i] != value) {
        i++;
    }

    if (SSNs[i] == value) {
        found = true;
        Console.WriteLine(names[i]);
    }
}

if (!found) {
    Console.WriteLine("This value does not exist");
}

```

39. Solution

```

const int STUDENTS = 12;
const int LESSONS = 6;

static void Main(string[] args) {

```



```

int i, j;
bool found, failure;

int[,] grades = new int[STUDENTS, LESSONS];
for (i = 0; i <= STUDENTS - 1; i++) {
    for (j = 0; j <= LESSONS - 1; j++) {
        do {
            grades[i, j] = Int32.Parse(Console.ReadLine());
            failure = false;
            if (grades[i, j] < 0) {
                Console.WriteLine("Error! You entered a negative value");
                failure = true;
            }
            else if (grades[i, j] > 100) {
                Console.WriteLine("Error! You entered a value grater than 100");
                failure = true;
            }
        } while (failure);
    }
}

double[] average = new double[STUDENTS];
for (i = 0; i <= STUDENTS - 1; i++) {
    average[i] = 0;
    for (j = 0; j <= LESSONS - 1; j++) {
        average[i] += grades[i, j];
    }
    average[i] /= LESSONS;
}

found = false;
for (i = 0; i <= STUDENTS - 1; i++) {
    if (average[i] < 70) {
        found = true;
        break;
    }
}

if (found) {
    Console.WriteLine("There is at least one student that has an average value below 70");
}
}

```

40. Solution

```

static void Main(string[] args) {
    string word, letter;
    int i;

    Dictionary<string, string> morseAlphabet = new Dictionary<string, string> {
        {"A", ".-"},
        {"B", "-..."},

```

```

        {"C", "-.-."},
        {"D", "-.."},
        {"E", "."},
        {"F", "-.-."},
        {"G", "--."},
        {"H", "..."},
        {"I", ".."},
        {"J", ".---"},
        {"K", "-.-"},
        {"L", "-.-."},
        {"M", "--"},
        {"N", "-."},
        {"O", "---"},
        {"P", "-.-."},
        {"Q", "--.-"},
        {"R", "-.-"},
        {"S", "..."},
        {"T", "-"},
        {"U", "..-"},
        {"V", "...-"},
        {"W", "-.-"},
        {"X", "-.-.-"},
        {"Y", "-.-.-"},
        {"Z", "--.."},
        {" ", "/" }
    };

    Console.Write("Enter a word: ");
    word = Console.ReadLine();

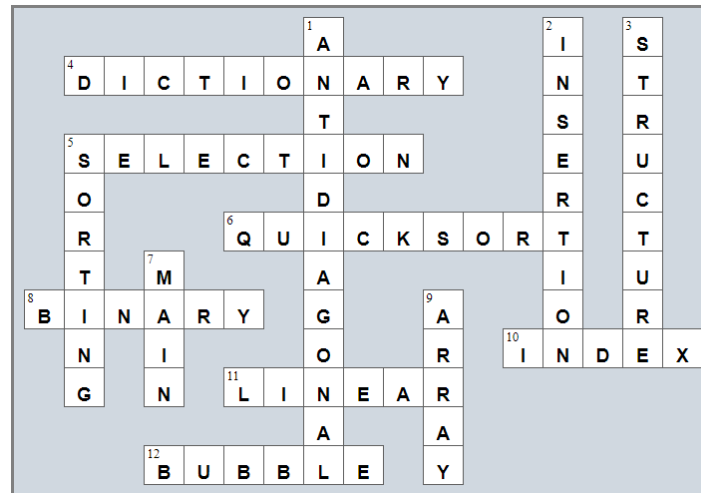
    for (i = 0; i <= word.Length - 1; i++) {
        letter = "" + word[i];
        Console.Write(morseAlphabet[letter.ToUpper()] + " ");
    }
}

```

Review in “Data Structures in C#”

Review Crossword Puzzle

1.



Chapter 35

35.4 Review Questions: True/False

- | | |
|----------|-----------|
| 1. true | 7. true |
| 2. true | 8. false |
| 3. false | 9. true |
| 4. false | 10. true |
| 5. true | 11. false |
| 6. true | 12. true |

Chapter 36

36.8 Review Questions: True/False

1. false

2. true

3. false

4. true

5. true

6. false

7. true

8. false

9. true

10. false

11. true

12. true

13. true

14. true

15. true

16. false

17. false
18. true

19. false

20. true

21. true

22. true

23. true

24. false

25. true

26. false

27. true

28. false

29. true

30. true

31. true

32. true

33. false

36.9 Review Exercises

1. Solution

```
static int find_max(int a, int b) {  
    int maximum;  
    if (a > b) {  
        maximum = a;  
    }  
    else {  
        maximum = b;  
    }  
    return maximum;  
}
```

2. Solution

Step	Statement	Main Code		Method sum_digits()		
		s	i	a	d1	d2
1	s = 0	0	?			
2	i = 25	0	25			
3	i <= 27	true				
4	s += sum_digits(i)			25	?	?
5	d1 = a % 10			25	5	?

6	d2 = (int)(a / 10)			25	5	2
7	return d1 + d2	7	25			
8	i++	7	26			
9	i <= 27	true				
10	s += sum_digits(i)			26	?	?
11	d1 = a % 10			26	6	?
12	d2 = (int)(a / 10)			26	6	2
13	return d1 + d2	15	26			
14	i++	15	27			
15	i <= 27	true				
16	s += sum_digits(i)			27	?	?
17	d1 = a % 10			27	7	?
18	d2 = (int)(a / 10)			27	7	2
19	return d1 + d2	24	27			
20	i++	24	28			
21	i <= 27	false				
22	Console.WriteLine(s)	It displays: 24				

3. Solution

Step	Statement	Main Code		Method sss()		
		s	i	a	total	k
1	i = 1	?	1			
2	s = 0	0	1			
3	while(i < 6)	true				
4	if (i % 2 == 1)	true				
5	s += 1	1	1			
6	i++	1	2			
7	while(i < 6)	true				
8	if (i % 2 == 1)	false				
9	s += sss(i)			2	?	?
10	total = 0			2	0	?
11	k = 1			2	0	1
12	k <= a			true		
13	total += k			2	1	1
14	k++			2	1	2
15	k <= a			true		
16	total += k			2	3	2

17	k++			2	3	3
18	k <= a			false		
19	return total	4	2			
20	i++	4	3			
21	while(i < 6)	true				
22	if (i % 2 == 1)	true				
23	s += 1	5	3			
24	i++	5	4			
25	while(i < 6)	true				
26	if (i % 2 == 1)	false				
27	s += sss(i)			4	?	?
28	total = 0			4	0	?
29	k = 1			4	0	1
30	k <= a			true		
31	total += k			4	1	1
32	k++			4	1	2
33	k <= a			true		
34	total += k			4	3	2
35	k++			4	3	3
36	k <= a			true		
37	total += k			4	6	4
38	k++			4	6	4
39	k <= a			true		
40	total += k			4	10	4
41	k++			4	10	5
42	k <= a			false		
43	return total	15	4			
44	i++	15	5			
45	while(i < 6)	true				
46	if (i % 2 == 1)	true				
47	s += 1	16	5			
48	i++	16	6			
49	while(i < 6)	false				
50	Console.WriteLine(s)	It displays: 16				

4. Solution

Step	Statement	Main Code				Method custom_div()	
		k	m	a	x	b	d
1	k = Int32.Parse(Console.ReadLine())	12	?	?	?		
2	m = 2	12	2	?	?		
3	a = 1	12	2	1	?		
4	while (a < 6)	true					
5	if (k % m != 0)	false					
6	x = a + m + custom_div(m, a)					2	1
7	return (int)((b + d) / 2)	12	2	1	4		
8	Console.WriteLine(m + " " + a + " " + x)	It displays: 2 1 4					
9	a += 2	12	2	3	4		
10	m++	12	3	3	4		
11	while (a < 6)	true					
12	if (k % m != 0)	false					
13	x = a + m + custom_div(m, a)					3	3
14	return (int)((b + d) / 2)	12	3	3	9		
15	Console.WriteLine(m + " " + a + " " + x)	It displays: 3 3 9					
16	a += 2	12	3	5	9		
17	m++	12	4	5	9		
18	while (a < 6)	true					
19	if (k % m != 0)	false					
20	x = a + m + custom_div(m, a)					4	5
21	return (int)((b + d) / 2)	12	4	5	13		
22	Console.WriteLine(m + " " + a + " " + x)	It displays: 4 5 13					
23	a += 2	12	4	7	13		
24	m++	12	5	7	13		
25	while (a < 6)	false					

5. Solution

Step	Statement	Main Code		void Method display()
		i	x	a
1	i = 1	1	?	
2	i <= 5	true		
3	x = Int32.Parse(Console.ReadLine())	1	3	

4	display(x)			3
5	if (a % 2 == 0)			false
6	Console.WriteLine(a + " is odd")	It displays: 3 is odd		
7	i++	2	3	
8	i <= 5	true		
9	x = Int32.Parse(Console.ReadLine())	2	7	
10	display(x)			7
11	if (a % 2 == 0)			false
12	Console.WriteLine(a + " is odd")	It displays: 7 is odd		
13	i++	3	7	
14	i <= 5	true		
15	x = Int32.Parse(Console.ReadLine())	3	9	
16	display(x)			9
17	if (a % 2 == 0)			false
18	Console.WriteLine(a + " is odd")	It displays: 9 is odd		
19	i++	4	9	
20	i <= 5	true		
21	x = Int32.Parse(Console.ReadLine())	4	2	
22	display(x)			2
23	if (a % 2 == 0)			true
24	Console.WriteLine(a + " is even")	It displays: 2 is even		
25	i++	5	2	
26	i <= 5	true		
27	x = Int32.Parse(Console.ReadLine())	5	4	
28	display(x)			4
29	if (a % 2 == 0)			true
30	Console.WriteLine(a + " is even")	It displays: 4 is even		
31	i++	6	4	
32	i <= 5	false		

6. Solution

Step	Statement	Main Code		void Method division()	
		x	y	a	b
1	x = 20	20	?		
2	y = 30	20	30		
3	while (x % y < 30)	true			

4	division(y, x)			30	20
5	b = (int)(b / a)			30	0
6	Console.WriteLine(a * b)	It displays: 0			
7	x = 4 * y	120	30		
8	y++	120	31		
9	while (x % y < 30)	true			
10	division(y, x)			31	120
11	b = (int)(b / a)			31	3
12	Console.WriteLine(a * b)	It displays: 93			
13	x = 4 * y	124	31		
14	y++	124	32		
15	while (x % y < 30)	true			
16	division(y, x)			32	124
17	b = (int)(b / a)			32	3
18	Console.WriteLine(a * b)	It displays: 96			
19	x = 4 * y	128	32		
20	y++	128	33		
21	while (x % y < 30)	true			
22	division(y, x)			33	128
23	b = (int)(b / a)			33	3
24	Console.WriteLine(a * b)	It displays: 99			
25	x = 4 * y	132	33		
26	y++	132	34		
27	while (x % y < 30)	false			

7. Solution

Step	Statement	Main Code		void Method calculate()		
		i	m	n	s	j
1	i = 1	1	?			
2	i <= 3	true				
3	m = Int32.Parse(Console.ReadLine())	1	2			
4	calculate(m)			2	?	?
5	s = 0			2	0	?
6	j = 2			2	0	2
7	j <= 2 * n			true		
8	s = s + Math.Pow(j, 2)			2	4	2

9	j += 2			2	4	4
10	j <= 2 * n			true		
11	s = s + Math.Pow(j, 2)			2	20	4
12	j += 2			2	20	6
13	j <= 2 * n			false		
14	Console.WriteLine(s)	It displays: 20				
15	i++	2	2			
16	i <= 3	true				
17	m = Int32.Parse(Console.ReadLine())	2	3			
18	calculate(m)			3	?	?
19	s = 0			3	0	?
20	j = 2			3	0	2
21	j <= 2 * n			true		
22	s = s + Math.Pow(j, 2)			3	4	2
23	j += 2			3	4	4
24	j <= 2 * n			true		
25	s = s + Math.Pow(j, 2)			3	20	4
26	j += 2			3	20	6
27	j <= 2 * n			true		
28	s = s + Math.Pow(j, 2)			3	56	6
29	j += 2			3	56	8
30	j <= 2 * n			false		
31	Console.WriteLine(s)	It displays: 56				
32	i++	3	3			
33	i <= 3	true				
34	m = Int32.Parse(Console.ReadLine())	3	4			
35	calculate(m)			4	?	?
36	s = 0			4	0	?
37	j = 2			4	0	2
38	j <= 2 * n			true		
39	s = s + Math.Pow(j, 2)			4	4	2
40	j += 2			4	4	4
41	j <= 2 * n			true		
42	s = s + Math.Pow(j, 2)			4	20	4
43	j += 2			4	20	6
44	j <= 2 * n			true		
45	s = s + Math.Pow(j, 2)			4	56	6

46	j += 2			4	56	8
47	j <= 2 * n			true		
48	s = s + Math.Pow(j, 2)			4	120	8
49	j += 2			4	120	10
50	j <= 2 * n			false		
51	Console.WriteLine(s)	It displays: 120				
52	i++	4	4			
53	i <= 3	false				

8. Solution

```
static double find_sum(double a, double b, double c) {
    return a + b + c;
}
```

9. Solution

```
static double find_avg(double a, double b, double c, double d) {
    return (a + b + c + d) / 4;
}
```

10. Solution

```
static double maximum(double a, double b, double c) {
    double m;

    m = a;
    if (b > m) {
        m = b;
    }
    if (c > m) {
        m = c;
    }
    return m;
}
```

11. Solution

```
static void display_max(double a, double b, double c, double d, double e) {
    double m;

    m = a;
    if (b > m) {
        m = b;
    }
    if (c > m) {
        m = c;
    }
    if (d > m) {
```

```
        m = d;
    }
    if (e > m) {
        m = e;
    }
    Console.WriteLine(m);
}
```

12. Solution

```
static double my_round(double x) {
    int digit_to_check;
    double return_value;

    digit_to_check = (int)(x * 1000) % 10;
    if (digit_to_check >= 5) {
        return_value = ((int)(x * 100) + 1) / 100.0;
    }
    else {
        return_value = ((int)(x * 100)) / 100.0;
    }

    return return_value;
}
```

13. Solution

```
static double find_min(double a, double b) {
    double minimum;

    minimum = a;
    if (b < minimum) {
        minimum = b;
    }
    return minimum;
}

static void Main(string[] args) {
    double temp1, temp2, x1, x2, x3, x4;

    Console.Write("Enter four numbers: ");
    x1 = Double.Parse(Console.ReadLine());
    x2 = Double.Parse(Console.ReadLine());
    x3 = Double.Parse(Console.ReadLine());
    x4 = Double.Parse(Console.ReadLine());

    //First approach
    temp1 = find_min(x1, x2);
    temp2 = find_min(x3, x4);
    Console.WriteLine(find_min(temp1, temp2));

    //Second approach
    Console.WriteLine(find_min(find_min(x1, x2), find_min(x3, x4)));
}
```

```

}

```

14. Solution

```

static double Kelvin_to_Fahrenheit(double kelvin) {
    return 1.8 * kelvin - 459.67;
}

static double Kelvin_to_Celsius(double kelvin) {
    return kelvin - 273.15;
}

static void Main(string[] args) {
    double k;

    Console.Write("Enter a temperature in degrees Kelvin: ");
    k = Double.Parse(Console.ReadLine());
    Console.WriteLine("Fahrenheit: " + Kelvin_to_Fahrenheit(k));
    Console.WriteLine("Celsius: " + Kelvin_to_Celsius(k));
}

```

15. Solution

```

static string bmi(double w, double h) {
    double b;
    string return_value;

    b = w * 703 / Math.Pow(h, 2);
    if (b < 16) {
        return_value = "You must add weight.";
    }
    else if (b < 18.5) {
        return_value = "You should add some weight.";
    }
    else if (b < 25) {
        return_value = "Maintain your weight.";
    }
    else if (b < 30) {
        return_value = "You should lose some weight.";
    }
    else {
        return_value = "You must lose weight.";
    }

    return return_value;
}

static void Main(string[] args) {
    double height, weight;
    int age;

    Console.Write("Enter your weight (in pounds): ");
    weight = Double.Parse(Console.ReadLine());
}

```

```
while (weight < 0) {
    Console.Write("Error! Enter your weight (in pounds): ");
    weight = Double.Parse(Console.ReadLine());
}

Console.WriteLine("Enter your age: ");
age = Int32.Parse(Console.ReadLine());
while (age < 18) {
    Console.Write("Error! Enter your age: ");
    age = Int32.Parse(Console.ReadLine());
}

Console.WriteLine("Enter your height (in inches): ");
height = Double.Parse(Console.ReadLine());
while (height < 0) {
    Console.WriteLine("Error! Enter your height (in inches): ");
    height = Double.Parse(Console.ReadLine());
}

Console.WriteLine(bmi(weight, height));
}
```

16. Solution

```
static void num_of_days(int year, int month) {
    int days;

    switch (month) {
        case 4:
        case 6:
        case 9:
        case 11:
            days = 30;
            break;
        case 2:
            if (year % 4 == 0 && year % 100 != 0 || year % 400 == 0) {
                days = 29;
            }
            else {
                days = 28;
            }
            break;
        default:
            days = 31;
            break;
    }

    Console.WriteLine(days);
}

static void Main(string[] args) {
    int m, y;
```

```

Console.Write("Enter a year: ");
y = Int32.Parse(Console.ReadLine());
for (m = 1; m <= 12; m++) {
    num_of_days(y, m);
}
}

```

17. Solution

```

static void display_menu() {
    Console.WriteLine();
    Console.WriteLine("1. Convert meters to miles");
    Console.WriteLine("2. Convert miles to meters");
    Console.WriteLine("3. Exit");
    Console.Write("Enter a choice: ");
}

static void meters_to_miles(double meters) {
    Console.WriteLine(meters + " meters equals " + (meters / 1609.344) + " miles");
}

static void miles_to_meters(double miles) {
    Console.WriteLine(miles + " miles equals " + (miles * 1609.344) + " meters");
}

static void Main(string[] args) {
    int choice;
    double distance;

    display_menu();
    choice = Int32.Parse(Console.ReadLine());
    while (choice != 3) {
        Console.WriteLine("Enter distance: ");
        distance = Int32.Parse(Console.ReadLine());
        if (choice == 1) {
            meters_to_miles(distance);
        }
        else {
            miles_to_meters(distance);
        }

        display_menu();
        choice = Int32.Parse(Console.ReadLine());
    }
}

```

18. Solution

```

static void amount_to_pay(int seconds) {
    double extra, tax, total, total_without_tax;

    if (seconds <= 600) {
        extra = 0;
    }
}

```



```
}  
else if (seconds <= 1200) {  
    extra = (seconds - 600) * 0.01;  
}  
else {  
    extra = 600 * 0.01 + (seconds - 1200) * 0.02;  
}  
  
total_without_tax = 10 + extra;  
tax = total_without_tax * 11 / 100;  
total = total_without_tax + tax;  
  
Console.WriteLine("Total amount to pay: " + total);  
}  
  
static void Main(string[] args) {  
    int seconds;  
  
    Console.Write("Enter number of seconds: ");  
    seconds = Int32.Parse(Console.ReadLine());  
    amount_to_pay(seconds);  
}
```

Chapter 37

37.10 Review Questions: True/False

- | | |
|-----------|-----------|
| 1. true | 14. false |
| 2. true | 15. true |
| 3. true | 16. true |
| 4. false | 17. false |
| 5. true | 18. false |
| 6. false | 19. false |
| 7. true | 20. false |
| 8. false | 21. true |
| 9. true | 22. true |
| 10. false | 23. false |
| 11. true | 24. true |
| 12. true | 25. true |
| 13. true | 26. true |

37.11 Review Exercises

1. Solution

It displays: 5

2. Solution

It displays: 14

3. Solution

It displays: 14

4. Solution

Step	Statement	Global		Main Code		void Method swap ()		
		arr[0]	arr[1]	k	x	x	y	temp
1	k = Int32.Parse(Console.ReadLine())	?	?	12	?			
2	arr[1] = 1	?	1	12	?			
3	arr[0] = 1	1	1	12	?			
4	while (arr[0] < 8)	1	1	true				
5	if (k % arr[1] != 0)	1	1	false				
6	x = arr[0] + arr[1] + (int)(arr[0] - arr[1])	1	1	12	2			
7	Console.WriteLine(arr[1] + " " + arr[0] + " " + x)	It displays: 1 1 2						
8	arr[0] += 2	3	1	12	2			
9	arr[1]++	3	2	12	2			

10	swap(arr[0], arr[1])					3	2	?
11	temp = x					3	2	3
12	x = y					2	2	3
13	y = temp					2	3	3
14	while (arr[0] < 8)	2	3	12	2			
		2	3	true				
15	if (k % arr[1] != 0)	2	3	false				
16	x = arr[0] + arr[1] + (int)(arr[0] - arr[1])	2	3	12	4			
17	Console.WriteLine(arr[1] + " " + arr[0] + " " + x)	It displays: 3 2 4						
18	arr[0] += 2	4	3	12	4			
19	arr[1]++	4	4	12	4			
20	swap(arr[0], arr[1])					4	4	?
21	temp = x					4	4	4
22	x = y					4	4	4
23	y = temp					4	4	4
24	while (arr[0] < 8)	4	4	12	4			
		4	4	true				
25	if (k % arr[1] != 0)	4	4	false				
26	x = arr[0] + arr[1] + (int)(arr[0] - arr[1])	4	4	12	8			
27	Console.WriteLine(arr[1] + " " + arr[0] + " " + x)	It displays: 4 4 8						
28	arr[0] += 2	6	4	12	8			
29	arr[1]++	6	5	12	8			
30	swap(arr[0], arr[1])					6	5	?
31	temp = x					6	5	6
32	x = y					5	5	6
33	y = temp					5	6	5
34	while (arr[0] < 8)	5	6	12	8			
		5	6	true				
35	if (k % arr[1] != 0)	5	6	false				
36	x = arr[0] + arr[1] + (int)(arr[0] - arr[1])	5	6	12	10			
37	Console.WriteLine(arr[1] + " " + arr[0] + " " + x)	It displays: 6 5 10						
38	arr[0] += 2	7	6	12	10			
39	arr[1]++	7	7	12	10			

40	swap(arr[0], arr[1])					7	7	?
41	temp = x					7	7	7
42	x = y					7	7	7
43	y = temp					7	7	7
44	while (arr[0] < 8)	7	7	12	10			
		7	7	true				
45	if (k % arr[1] != 0)	7	7	true				
46	x = arr[0] % arr[1]	7	7	12	0			
47	swap(arr[1], arr[0])					7	7	?
48	temp = x					7	7	7
49	x = y					7	7	7
50	y = temp					7	7	7
51	Console.WriteLine(arr[1] + " " + arr[0] + " " + x)	7	7	12	0			
		It displays: 7 7 0						
52	arr[0] += 2	9	7	12	0			
53	arr[1]++	9	8	12	0			
54	swap(arr[0], arr[1])					9	8	?
55	temp = x					9	8	9
56	x = y					8	8	9
57	y = temp					8	9	9
58	while (arr[0] < 8)	8	9	12	0			
		false						

5. Solution

It displays: hellohellohello

6. Solution

It displays: 15

7. Solution

It displays: 11 4

8. Solution

```
const int STUDENTS = 10;
const int LESSONS = 5;

static void part1(string[] names, int[,] grades) {
    int i, j;

    for (i = 0; i <= STUDENTS - 1; i++) {
        Console.Write("Enter name for student No. " + (i + 1) + ": ");
```

```

    names[i] = Console.ReadLine();
    for (j = 0; j <= LESSONS - 1; j++) {
        Console.WriteLine("Enter grade for lesson No. " + (j + 1) + ": ");
        grades[i, j] = Int32.Parse(Console.ReadLine());
    }
}
}

static double[] part2(int[,] grades) {
    double[] average = new double[STUDENTS];
    int i, j;

    for (i = 0; i <= STUDENTS - 1; i++) {
        average[i] = 0;
        for (j = 0; j <= LESSONS - 1; j++) {
            average[i] += grades[i, j];
        }
        average[i] /= LESSONS;
    }
    return average;
}

static void part3(double[] average, string[] names) {
    int m, n;
    double temp;
    string temp_str;

    for (m = 1; m <= STUDENTS - 1; m++) {
        for (n = STUDENTS - 1; n >= m; n--) {
            if (average[n] > average[n - 1]) {
                temp = average[n];
                average[n] = average[n - 1];
                average[n - 1] = temp;

                temp_str = names[n];
                names[n] = names[n - 1];
                names[n - 1] = temp_str;
            }
            else if (average[n] == average[n - 1]) {
                if (names[n].CompareTo(names[n - 1]) < 0) {
                    temp_str = names[n];
                    names[n] = names[n - 1];
                    names[n - 1] = temp_str;
                }
            }
        }
    }
}

static void Main(string[] args) {
    int i;

    string[] names = new string[STUDENTS];

```

```

int[,] grades = new int[STUDENTS, LESSONS];
double[] average = new double[STUDENTS];

part1(names, grades);

average = part2(grades);

part3(average, names);

for (i = 0; i <= STUDENTS - 1; i++) {
    Console.WriteLine(names[i] + "\t" + average[i]);
}
}

```

9. Solution

```

static string part1() {
    string message;

    Console.Write("Enter a message: ");
    message = Console.ReadLine().ToLower();
    return message;
}

static string part2(string message) {
    string letter, message_clean;
    int i;

    message_clean = "";
    for (i = 0; i <= message.Length - 1; i++) {
        letter = "" + message[i];
        if (letter != " " && letter != "," && letter != "." && letter != "?") {
            message_clean += letter;
        }
    }
    return message_clean;
}

static bool part3(string message_clean) {
    int middle_pos, i, j;
    bool palindrome;
    string left_letter, right_letter;

    middle_pos = (int)(message_clean.Length - 1) / 2;
    j = message_clean.Length - 1;
    palindrome = true;
    for (i = 0; i <= middle_pos; i++) {
        left_letter = "" + message_clean[i];
        right_letter = "" + message_clean[j];
        if (left_letter != right_letter) {
            palindrome = false;
            break;
        }
    }
}

```

```

        j--;
    }
    return palindrome;
}

static bool part4(string message) {
    string message_clean;
    bool palindrome;

    message_clean = part2(message);
    palindrome = part3(message_clean);
    return palindrome;
}

static void Main(string[] args) {
    string message;
    bool palindrome;

    message = part1();
    palindrome = part4(message);
    if (palindrome) {
        Console.WriteLine("The message is palindrome");
    }
}

```

10. Solution

```

static void Main(string[] args) {
    int a, b, c, maximum;

    a = Int32.Parse(Console.ReadLine());
    b = Int32.Parse(Console.ReadLine());
    c = Int32.Parse(Console.ReadLine());
    d = Int32.Parse(Console.ReadLine());

    maximum = a;
    if (b > maximum) {
        maximum = b;
    }
    if (c > maximum) {
        maximum = c;
    }
    if (d > maximum) {
        maximum = d;
    }

    Console.WriteLine(maximum);
}

```

11. Solution

```

static void f1(double a, double b, double c, double[] returning_array) {
    returning_array[0] = a + b + c;
}

```

```

    returning_array[1] = returning_array[0] = / 3;
}

```

12. Solution

```

static double my_round(double x, int decimal_places = 2) {
    double return_value;

    int digit_to_check = (int)((x * Math.Pow(10, decimal_places + 1)) % 10);
    if (digit_to_check >= 5) {
        return_value = ((int)((x * Math.Pow(10, decimal_places))) + 1) / Math.Pow(10, decimal_places);
    }
    else {
        return_value = ((int)(x * Math.Pow(10, decimal_places))) / Math.Pow(10, decimal_places);
    }
    return return_value;
}

```

13. Solution

```

static bool get_input() {
    string answer;

    do {
        Console.Write("Enter Yes or No: ");
        answer = Console.ReadLine().ToUpper();
    } while (answer != "YES" && answer != "NO");

    return answer == "YES"; //This returns true or false
}

static double find_area(double b, double h) {
    return b * h;
}

static void Main(string[] args) {
    double b, h;

    do {
        Console.Write("Enter the base of the parallelogram: ");
        b = Double.Parse(Console.ReadLine());
        Console.Write("Enter the height of the parallelogram: ");
        h = Double.Parse(Console.ReadLine());

        Console.WriteLine("Area = " + find_area(b, h));

        Console.WriteLine("Would you like to repeat? ");
    } while (get_input());
}

```

14. Solution

```

const int STUDENTS = 100;

```



```
static void get_arrays(string[] names, int[] grades) {
    int i;

    for (i = 0; i <= STUDENTS - 1; i++) {
        Console.Write("Enter name: ");
        names[i] = Console.ReadLine();
        Console.Write("Enter grade: ");
        grades[i] = Int32.Parse(Console.ReadLine());
    }
}

static double get_average(int[] grades) {
    int i, total = 0;
    for (i = 0; i <= STUDENTS - 1; i++) {
        total += grades[i];
    }
    return total / (double)STUDENTS;
}

static void sort_arrays(int[] grades, string[] names) {
    int m, n, element_grds;
    string element_nms;

    for (m = 1; m <= STUDENTS - 1; m++) {
        element_grds = grades[m];
        element_nms = names[m];

        n = m;
        while (n > 0 && grades[n - 1] > element_grds) {
            grades[n] = grades[n - 1];
            names[n] = names[n - 1];
            n--;
        }

        grades[n] = element_grds;
        names[n] = element_nms;
    }
}

static void Main(string[] args) {
    int i;
    double average;

    string[] names = new string[STUDENTS];
    int[] grades = new int[STUDENTS];

    get_arrays(names, grades);
    average = get_average(grades);
    sort_arrays(grades, names);
    for (i = 0; i <= STUDENTS - 1; i++) {
        if (grades[i] < average) {
            Console.WriteLine(names[i]);
        }
    }
}
```

```

    }
}

```

15. Solution

```

const int JUDGES = 10;

static int[] get_array() {
    int[] score = new int[JUDGES];
    int i;

    for (i = 0; i <= JUDGES - 1; i++) {
        Console.Write("Judge No " + (i + 1) + ". Enter score: ");
        score[i] = Int32.Parse(Console.ReadLine());
    }
    return score;
}

static void find_min_max(int[] score, ref int minimum, ref int maximum) {
    int i;
    minimum = score[0];
    maximum = score[0];
    for (i = 1; i <= JUDGES - 1; i++) {
        if (score[i] > maximum) {
            maximum = score[i];
        }
        if (score[i] < minimum) {
            minimum = score[i];
        }
    }
}

static void Main(string[] args) {
    string name;
    int total, i, points, minimum = 0, maximum = 0;

    Console.Write("Enter artist's name: ");
    name = Console.ReadLine();
    int[] score = get_array();
    find_min_max(score, ref minimum, ref maximum);

    total = 0;
    for (i = 0; i <= JUDGES - 1; i++) {
        total += score[i];
    }

    points = total - minimum - maximum;
    Console.WriteLine("Artist " + name + " got " + points + " points");
}

```

16. Solution

```

static double woc(int index) {

```

```
double return_value;

if (index == 1) {
    return_value = 1;
}
else {
    return_value = 2 * woc(index - 1);
}
return return_value;
}

static void Main(string[] args) {
    double total;
    int i;

    total = 0;
    for (i = 1; i <= 64; i++) {
        total += woc(i);
    }
    Console.WriteLine(total);
}
```

17. Solution

```
static double factorial(int value) {
    double return_value;

    if (value == 1) {
        return_value = 1;
    }
    else {
        return_value = value * factorial(value - 1);
    }

    return return_value;
}

static double my_cos(double x, int i = 40) {
    double return_value;

    if (i == 0) {
        return_value = 1;
    }
    else {
        return_value = my_cos(x, i - 4) + Math.Pow(x, i) / factorial(i)
                        - Math.Pow(x, i - 2) / factorial(i - 2);
    }

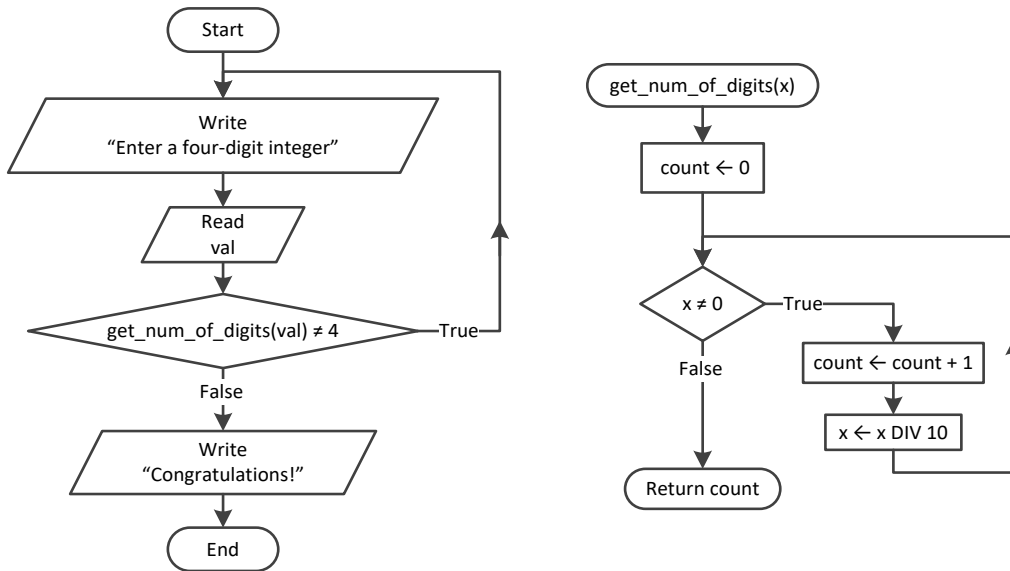
    return return_value;
}

static void Main(string[] args) {
    Console.WriteLine(my_cos(Math.PI / 4));
}
```

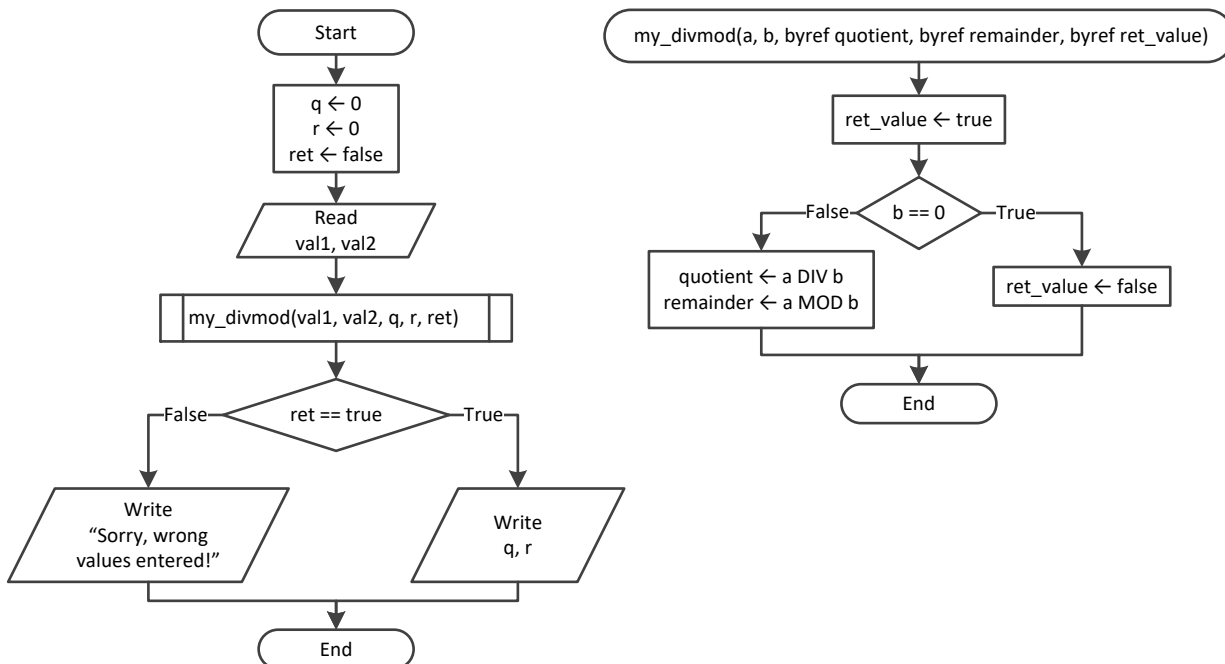
Chapter 38

38.3 Review Exercises

1. Solution



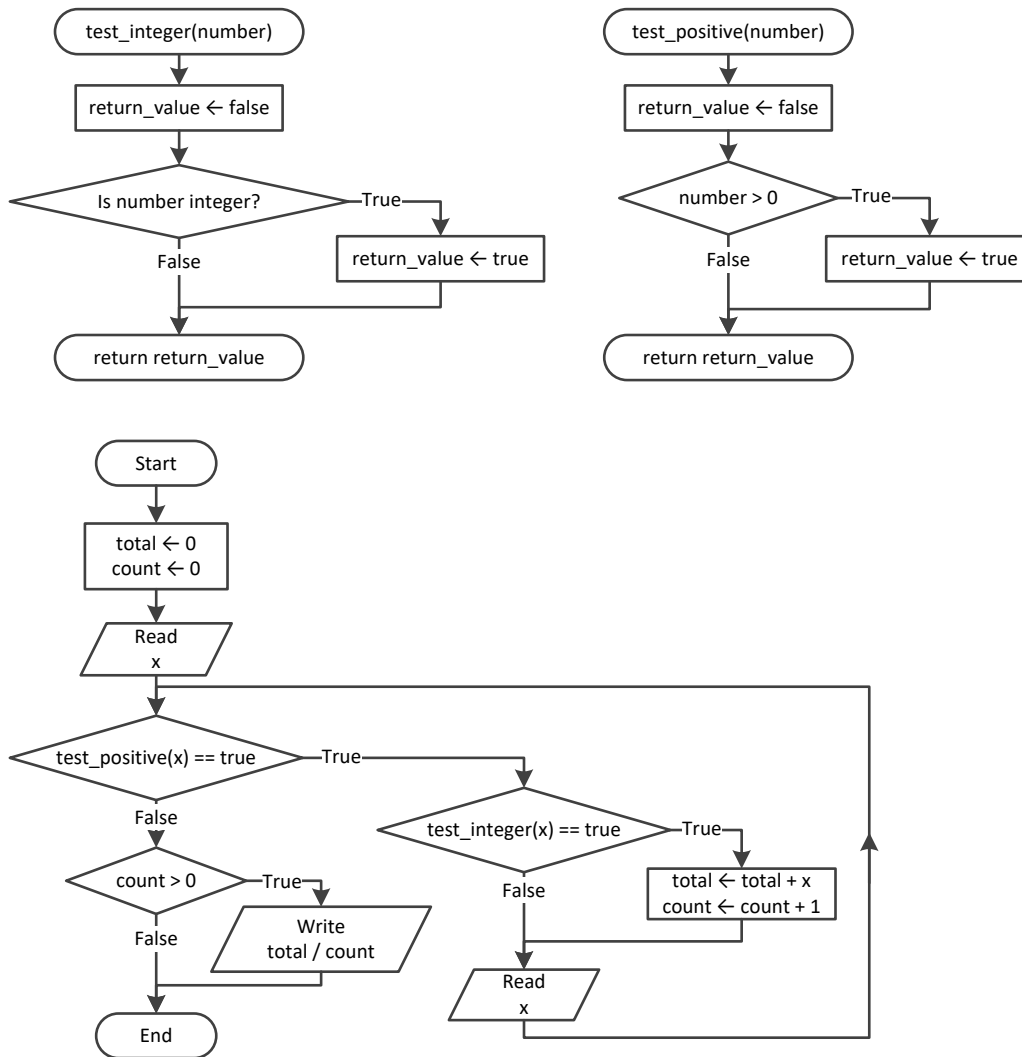
2. Solution



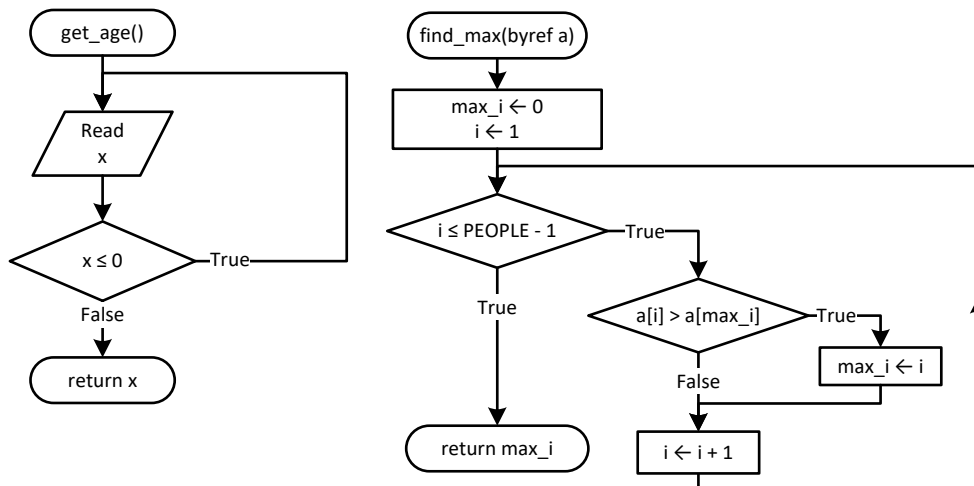
Flowcharts are a loose method of representing an algorithm. Thus, you can represent a pass by reference using the keyword `byref`, which clearly denotes what it actually does.

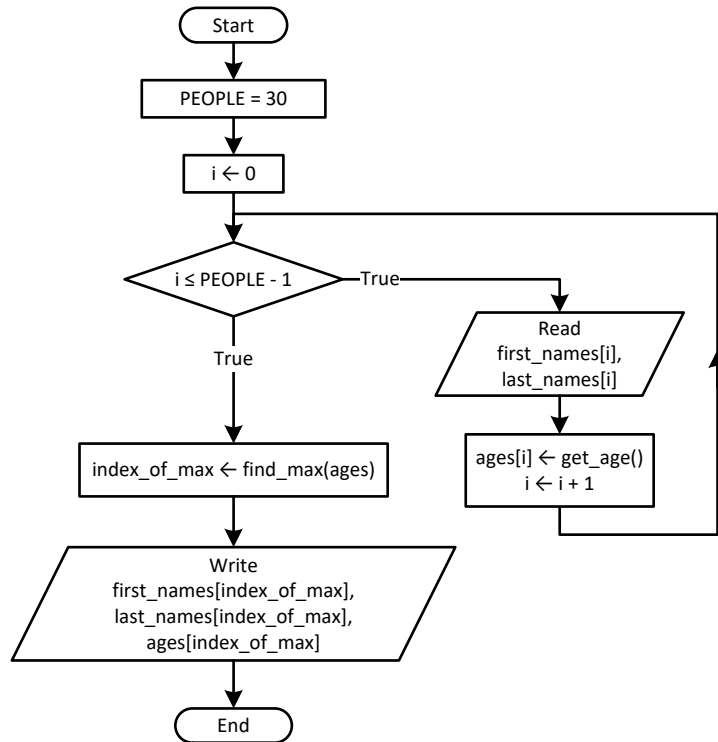
Some programmers, instead of using the keyword `byref`, prefer to write the keyword `inout`, which denotes pretty much the same thing— that the variable is both input (it accepts values) and output (it returns values).

3. Solution

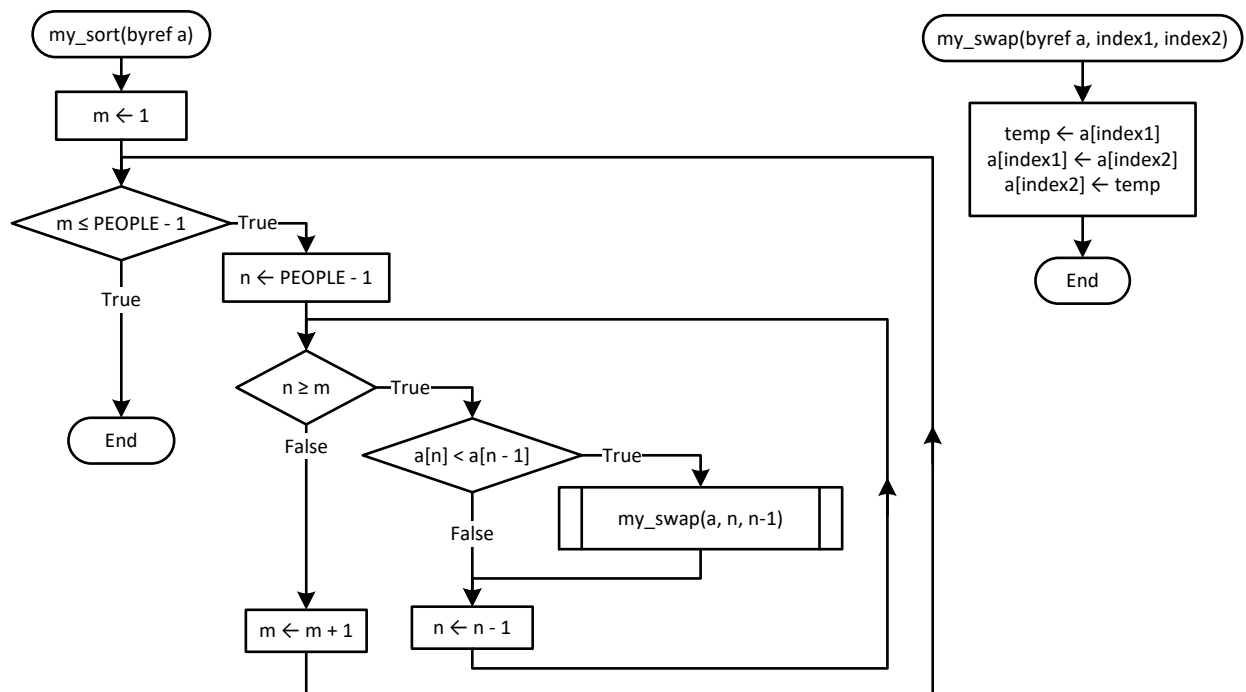


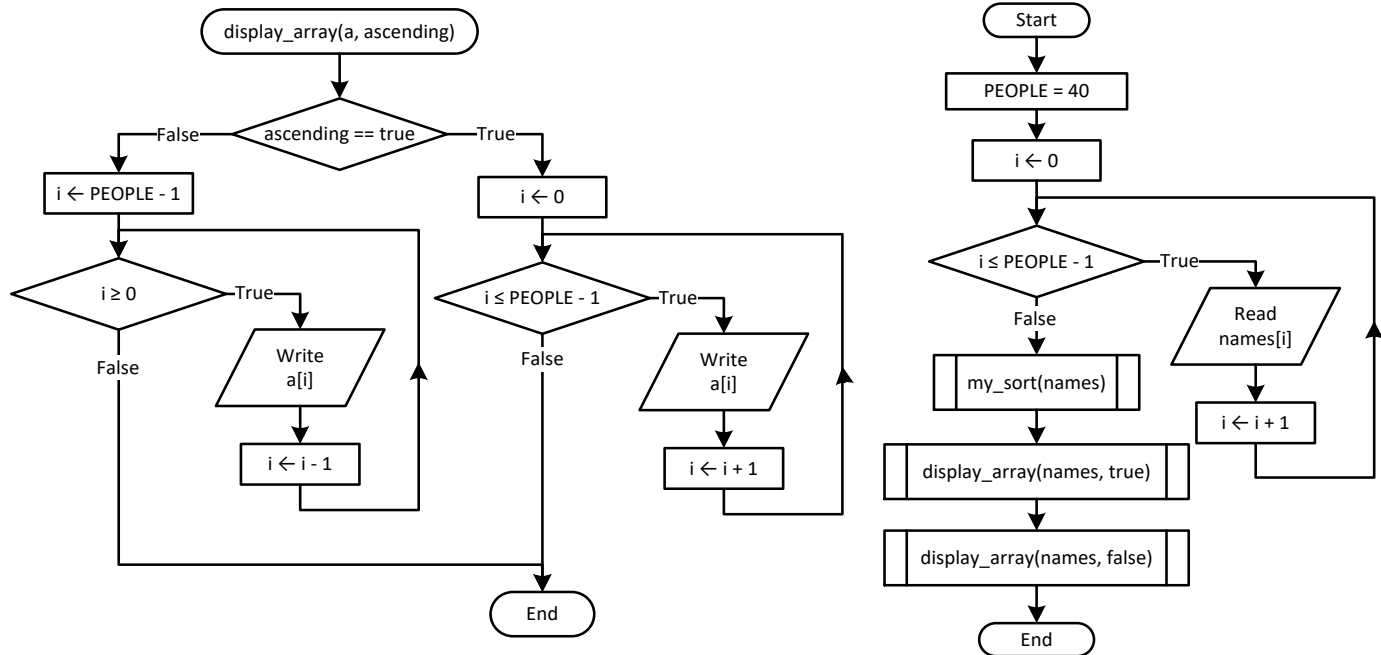
4. Solution





5. Solution





6. Solution

```

static double f1(int n) {
    double s;
    int i;

    s = 0;
    for (i = 1; i <= n; i++) {
        if (i < n / 2.0) {
            s += Math.Pow(n, 2);
        }
        else {
            s += Math.Pow(n, 3);
        }
    }
    return s;
}

static void Main(string[] args) {
    int val;

    do {
        Console.Write("Enter a positive integer ");
        val = Int32.Parse(Console.ReadLine());
    } while (val < 0);
    Console.WriteLine(f1(val));
}

```

7. Solution

```

const int ELEMENTS = 100;

```

```

static double[] read_values() {
    double[] values = new double[ELEMENTS];
    int i;

    for (i = 0; i <= ELEMENTS - 1; i++) {
        values[i] = Int32.Parse(Console.ReadLine());
    }
    return values;
}


static void find_min_max(double[] values, ref int min_i, ref int max_i) {
    int i;

    min_i = max_i = 0;
    for (i = 1; i <= ELEMENTS - 1; i++) {
        if (values[i] < values[min_i]) {
            min_i = i;
        }
        if (values[i] > values[max_i]) {
            max_i = i;
        }
    }
}

static void Main(string[] args) {
    double[] v = new double[ELEMENTS];
    int min_i = 0, max_i = 0;

    v = read_values();
    find_min_max(v, ref min_i, ref max_i);
    Console.WriteLine(v[min_i] + ", " + v[max_i]);
}

```

 Please note the way the void method `find_min_max()` finds the index positions of the minimum and the maximum values of the array values. This method is not the same as the one you learned in paragraph 34.3; however, it can be used as an alternative.

8. Solution

```

const double ACCURACY = 0.000000001;

static double factorial(int n) {
    int i;

    double return_value = 1;
    for (i = 1; i <= n; i++) {
        return_value *= i;
    }
    return return_value;
}

static double my_sin(double x) {
    int i, sign;

```



```

double sinus, sinus_previous;
sign = 1;
sinus = 0;
i = 1;
do {
    sinus_previous = sinus;
    sinus += sign * Math.Pow(x, i) / factorial(i);

    sign = -sign;
    i += 2;
} while (Math.Abs(sinus - sinus_previous) > ACCURACY);
return sinus;
}

static double degrees_to_rad(double degrees) {
    return 2 * Math.PI * degrees / 360;
}

static void Main(string[] args) {
    int i;

    for (i = 0; i <= 360; i++) {
        Console.WriteLine("sin(" + i + ") ~= " + my_sin(degrees_to_rad(i)));
    }
}

```

9. Solution

```

static bool is_leap(int year) {
    bool return_value = false;
    if (year % 4 == 0 && year % 100 != 0 || year % 400 == 0) {
        return_value = true;
    }
    return return_value;
}

static int num_of_days(int year, int month) {
    int days;

    switch (month) {
        case 4:
        case 6:
        case 9:
        case 11:
            days = 30;
            break;
        case 2:
            if (is_leap(year)) {
                days = 29;
            }
            else {
                days = 28;
            }
    }
}

```

```

        break;
    default:
        days = 31;
        break;
    }

    return days;
}

static bool check_date(int day, int month, int year) {
    bool return_value = true;
    if (month < 1 || month > 12) {
        return_value = false;
    }
    else if (day < 1 || day > num_of_days(year, month)) {
        return_value = false;
    }
    return return_value;
}

static void Main(string[] args) {
    int day, month, year, total, i;

    Console.Write("Enter day: ");
    day = Int32.Parse(Console.ReadLine());
    Console.Write("Enter month: ");
    month = Int32.Parse(Console.ReadLine());
    Console.Write("Enter year: ");
    year = Int32.Parse(Console.ReadLine());
    while (!check_date(day, month, year)) {
        Console.WriteLine("Error!");
        Console.Write("Enter day: ");
        day = Int32.Parse(Console.ReadLine());
        Console.Write("Enter month: ");
        month = Int32.Parse(Console.ReadLine());
        Console.Write("Enter year: ");
        year = Int32.Parse(Console.ReadLine());
    }

    total = 0;
    for (i = 1; i <= month - 1; i++) {
        total += num_of_days(year, i);
    }
    total += day;

    Console.WriteLine(total);
}

```

10. Solution

```

static void display_menu() {
    Console.WriteLine("-----");
    Console.WriteLine("1. Convert USD to Euro (EUR)");
}

```

```
Console.WriteLine("2. Convert USD to British Pound Sterling (GBP)");
Console.WriteLine("3. Convert EUR to USD");
Console.WriteLine("4. Convert EUR to GBP");
Console.WriteLine("5. Convert GBP to USD");
Console.WriteLine("6. Convert GBP to EUR");
Console.WriteLine("7. Exit");
Console.WriteLine("-----");
Console.Write("Enter a choice: ");
}

static double USD_to_EUR(double value) {
    return value * 0.87;
}

static double USD_to_GBP(double value) {
    return value * 0.76;
}

static void Main(string[] args) {
    int choice;
    double amount;

    display_menu();
    choice = Int32.Parse(Console.ReadLine());
    while (choice != 7) {
        Console.Write("Enter an amount: ");
        amount = Double.Parse(Console.ReadLine());
        switch (choice) {
            case 1:
                Console.WriteLine(amount + " USD = " + USD_to_EUR(amount) + " Euro");
                break;
            case 2:
                Console.WriteLine(amount + " USD = " + USD_to_GBP(amount) + " GBP");
                break;
            case 3:
                Console.WriteLine(amount + " EUR = " + 1 / USD_to_EUR(1 / amount) + " USD");
                break;
            case 4:
                Console.WriteLine(amount + " EUR = " + USD_to_GBP(1 / USD_to_EUR(1 / amount)) + " GBP");
                break;
            case 5:
                Console.WriteLine(amount + " GBP = " + 1 / USD_to_GBP(1 / amount) + " USD");
                break;
            case 6:
                Console.WriteLine(amount + " GBP = " + USD_to_EUR(1 / USD_to_GBP(1 / amount)) + " EUR");
                break;
        }

        display_menu();
        choice = Int32.Parse(Console.ReadLine());
    }
}
```

11. Solution

```
static int dice() {
    Random rnd = new Random();
    return rnd.Next(1, 7);
}

static void Main(string[] args) {
    int dice1, dice2, i, player, total, total_player1 = 0, total_player2 = 0;
    string[] names = new string[2];

    Console.Write("Player1 - Enter name: ");
    names[0] = Console.ReadLine();
    Console.Write("Player2 - Enter name: ");
    names[1] = Console.ReadLine();

    for (player = 0; player <= 1; player++) {
        total = 0;
        for (i = 1; i <= 10; i++) {
            Console.WriteLine(names[player] + ", hit enter to roll the dice!");
            Console.ReadLine(); //This statement just waits the user to hit the enter key

            dice1 = dice();
            dice2 = dice();
            Console.WriteLine(dice1 + "    " + dice2);
            total += dice1 + dice2;
        }
        if (player == 1) {
            total_player1 = total;
        }
        else {
            total_player2 = total;
        }
    }

    if (total_player1 == total_player2) {
        Console.WriteLine("Tie!");
    }
    else if (total_player1 > total_player2) {
        Console.WriteLine(names[0] + " wins");
    }
    else {
        Console.WriteLine(names[1] + " wins");
    }
}
```

12. Solution

```
const int GAS = 1;
const int DIESEL = 2;
const int HYBRID = 3;
const double TAX_RATE = 0.10;
```

```
const int CARS = 40;

static int get_choice() {
    Console.WriteLine("1. Gas");
    Console.WriteLine("2. Diesel");
    Console.WriteLine("3. Hybrid");
    Console.Write("Enter type of the car: ");
    return Int32.Parse(Console.ReadLine());
}

static int get_days() {
    Console.Write("Enter total number of rental days: ");
    return Int32.Parse(Console.ReadLine());
}

static double get_charge(int car_type, int rental_days) {
    double charge;

    if (car_type == GAS) {
        if (rental_days <= 5) {
            charge = rental_days * 24;
        }
        else if (rental_days <= 8) {
            charge = 5 * 24 + (rental_days - 5) * 22;
        }
        else {
            charge = 5 * 24 + 3 * 22 + (rental_days - 8) * 18;
        }
    }
    else if (car_type == DIESEL) {
        if (rental_days <= 5) {
            charge = rental_days * 28;
        }
        else if (rental_days <= 8) {
            charge = 5 * 28 + (rental_days - 5) * 25;
        }
        else {
            charge = 5 * 28 + 3 * 25 + (rental_days - 8) * 21;
        }
    }
    else {
        if (rental_days <= 5) {
            charge = rental_days * 30;
        }
        else if (rental_days <= 8) {
            charge = 5 * 30 + (rental_days - 5) * 28;
        }
        else {
            charge = 5 * 30 + 3 * 28 + (rental_days - 8) * 23;
        }
    }
    charge = charge * (1 + TAX_RATE); //This is equivalent to charge += charge * TAX_RATE;
```

```

    return charge;
}

static void Main(string[] args) {
    int count, i;
    double charge, total;

    int[] rented_car_types = new int[CARS];
    int[] rented_days = new int[CARS];

    for (i = 0; i <= CARS - 1; i++) {
        rented_car_types[i] = get_choice();
        rented_days[i] = get_days();
    }

    total = 0;
    for (i = 0; i <= CARS - 1; i++) {
        charge = get_charge(rented_car_types[i], rented_days[i]);
        Console.WriteLine("Car No " + (i + 1) + ": " + charge);
        total += charge;
    }

    count = 0;
    for (i = 0; i <= CARS - 1; i++) {
        if (rented_car_types[i] == HYBRID) {
            count++;
        }
    }

    Console.WriteLine("Hybrids rented: " + count);
    Console.WriteLine("Net profit: " + total / (1 + TAX_RATE));
}

```

13. Solution

```

const int CHANNELS = 10;
const int DAYS = 7;
const string day_names[] = {"Monday", "Tuesday", "Wednesday",
                             "Thursday", "Friday", "Saturday", "Sunday"};

static void get_data(string[] names, int[,] viewers) {
    int i, j;

    for (i = 0; i <= CHANNELS - 1; i++) {
        Console.WriteLine("Enter name for channel No. " + (i + 1) + ": ");
        names[i] = Console.ReadLine();
        for (j = 0; j <= DAYS - 1; j++) {
            Console.Write("Enter the number of viewers of the main news program on " + day_names[j]);
            Console.Write(" for channel " + names[i] + ": ");
            viewers[i, j] = Int32.Parse(Console.ReadLine());
        }
    }
}

```

```

static double get_average(int a[]) {
    int total ,i;

    total = 0;
    for (i = 0; i <= 4; i++) {
        total += a[i];
    }
    return total / 5.0;
}

static void Main(string[] args) {
    int i, j;
    double weekend;
    bool increasing;

    string[] names = new string[CHANNELS];
    int[,] viewers = new int[CHANNELS, DAYS];
    get_data(names, viewers);

    int[] temporary_array = new int[5];
    for (i = 0; i <= CHANNELS - 1; i++) {
        for (j = 0; j <= 4; j++) {
            temporary_array[j] = viewers[i, j];
        }
        weekend = (viewers[i, DAYS - 2] + viewers[i, DAYS - 1]) / 2;
        if (weekend >= 1.2 * get_average(temporary_array)) {
            Console.WriteLine(names[i]);
        }
    }

    for (i = 0; i <= CHANNELS - 1; i++) {
        increasing = true;
        for (j = 1; j <= DAYS - 1; j++) {
            if (viewers[i, j] <= viewers[i, j - 1]) {
                increasing = false;
            }
        }
        if (increasing) {
            Console.WriteLine(names[i]);
        }
    }
}

```

14. Solution

```

const int CITIZENS = 300;

static void input_data(long[] SSNs, string[] answers) {
    int i;

    for (i = 0; i <= CITIZENS - 1; i++) {
        Console.Write("Enter SSN: ");
    }
}

```

```

        SSNs[i] = Int64.Parse(Console.ReadLine());
        Console.Write("Enter answer: ");
        answers[i] = Console.ReadLine();
    }
}

static void sort_arrays(long[] SSNs, string[] answers) {
    int m, n, index_of_min;
    long minimum, temp;
    string temp_str;

    for (m = 0; m <= CITIZENS - 1; m++) {
        minimum = SSNs[m];
        index_of_min = m;
        for (n = m; n <= CITIZENS - 1; n++) {
            if (SSNs[n] < minimum) {
                minimum = SSNs[n];
                index_of_min = n;
            }
        }
        temp = SSNs[m];
        SSNs[m] = SSNs[index_of_min];
        SSNs[index_of_min] = temp;
        temp_str = answers[m];
        answers[m] = answers[index_of_min];
        answers[index_of_min] = temp_str;
    }
}

static int search_array(long[] SSNs, long SSN) {
    int left, right, middle, index_position = 0, return_value;
    bool found;

    left = 0;
    right = CITIZENS - 1;
    found = false;
    while (left <= right && !found) {
        middle = (int)((left + right) / 2);

        if (SSNs[middle] > SSN) {
            right = middle - 1;
        }
        else if (SSNs[middle] < SSN) {
            left = middle + 1;
        }
        else {
            found = true;
            index_position = middle;
        }
    }

    if (!found) {

```



```

        Console.WriteLine("SSN not found!");
        return_value = -1;
    }
    else {
        return_value = index_position;
    }
    return return_value;
}

static int count_answers(string[] answers, string answer) {
    int count, i;

    count = 0;
    for (i = 0; i <= CITIZENS - 1; i++) {
        if (answers[i] == answer) {
            count++;
        }
    }
    return count;
}

static void Main(string[] args) {
    long[] SSNs = new long[CITIZENS];
    long SSN;
    string[] answers = new string[CITIZENS];
    int index, count;
    string answer;

    do {
        input_data(SSNs, answers);
        sort_arrays(SSNs, answers);

        Console.Write("Enter an SSN to search: ");
        SSN = Int64.Parse(Console.ReadLine());

        index = search_array(SSNs, SSN);
        if (index != -1) {
            answer = answers[index];
            Console.WriteLine(answer);

            count = count_answers(answers, answer);
            Console.WriteLine(count * 100 / (double)CITIZENS);
        }
        Console.WriteLine("Repeat? ");
        answer = Console.ReadLine();
    } while (answer == "yes");
}

```

15. Solution

```

const int TEAMS = 8;
const int GAMES = 12;

```

```

static void input_data(string[] names, string[,] results) {
    int i, j;

    for (i = 0; i <= TEAMS - 1; i++) {
        Console.Write("Enter team name: ");
        names[i] = Console.ReadLine();
        for (j = 0; j <= GAMES - 1; j++) {
            Console.Write("Enter result (W, L, T): ");
            results[i, j] = Console.ReadLine();
        }
    }
}

static void display_result(string[] names, string[,] results) {
    string result;
    int i, j;
    bool found;

    Console.Write("Enter a result to search (W, L, T): ");
    result = Console.ReadLine();
    for (i = 0; i <= TEAMS - 1; i++) {
        Console.WriteLine("Team: " + names[i]);
        found = false;
        for (j = 0; j <= GAMES - 1; j++) {
            if (results[i, j] == result) {
                Console.WriteLine("Week: " + (j + 1));
                found = true;
            }
        }
        if (!found) {
            Console.WriteLine("Nothing found");
        }
    }
}

static int find_team(string[] names) {
    string name;
    int i, return_value;

    Console.Write("Enter a name to search: ");
    name = Console.ReadLine();

    i = 0;
    while (i < TEAMS - 1 && names[i] != name) {
        i++;
    }

    if (names[i] != name) {
        return_value = -1;
    }
    else {
        return_value = i;
    }
}

```

```
    return return_value;
}

static void Main(string[] args) {
    string[] names = new string[TEAMS];
    string[,] results = new string[TEAMS, GAMES];
    int j, index, total;

    input_data(names, results);
    display_result(names, results);

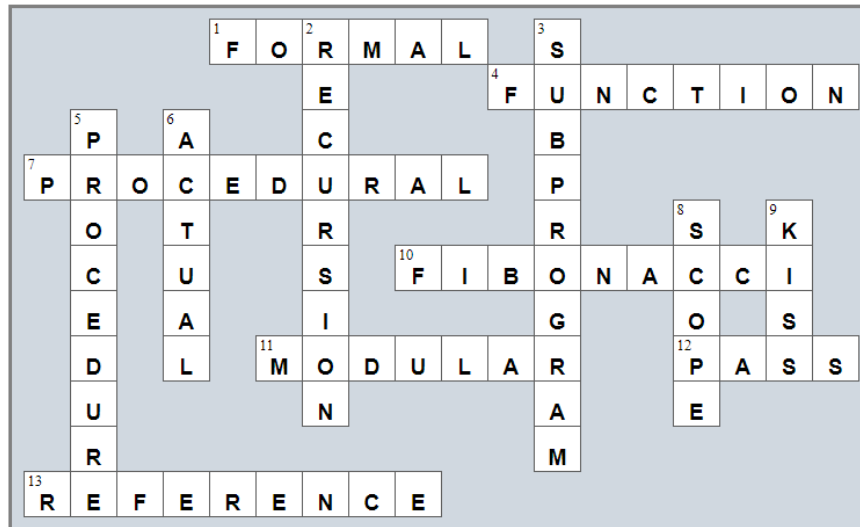
    index = find_team(names);
    while (index != -1) {
        total = 0;
        for (j = 0; j <= GAMES - 1; j++) {
            if (results[index, j] == "W") {
                total += 3;
            }
            else if (results[index, j] == "T") {
                total += 1;
            }
        }
        Console.WriteLine("Points: " + total);
        index = find_team(names);
    }

    if (index == -1) {
        Console.WriteLine("Team not found");
    }
}
```

Review in “Subprograms”

Review Crossword Puzzle

1.



Chapter 39

39.9 Review Questions: True/False

- | | | |
|----------|-----------|-----------|
| 1. false | 7. false | 13. true |
| 2. true | 8. true | 14. false |
| 3. true | 9. true | 15. true |
| 4. false | 10. false | 16. false |
| 5. false | 11. true | 17. false |
| 6. false | 12. true | |

39.10 Review Exercises

1. Solution

```
class Trigonometry {
    public double square_area(double side) {
        return side * side;
    }

    public double rectangle_area(double b, double h) {
        return b * h;
    }

    public double triangle_area(double b, double h) {
        return b * h / 2;
    }
}

class Program {
    static void Main(string[] args) {
        double sqr_side, rctngl_base, rctngl_height, trngl_base, trngl_height;
        Trigonometry tr = new Trigonometry();

        Console.Write("Enter square side: ");
        sqr_side = Double.Parse(Console.ReadLine());

        Console.Write("Enter rectangle base: ");
        rctngl_base = Double.Parse(Console.ReadLine());
        Console.Write("Enter rectangle height: ");
        rctngl_height = Double.Parse(Console.ReadLine());

        Console.Write("Enter triangle base: ");
        trngl_base = Double.Parse(Console.ReadLine());
        Console.Write("Enter triangle height: ");
        trngl_height = Double.Parse(Console.ReadLine());

        Console.WriteLine(tr.square_area(sqr_side));
        Console.WriteLine(tr.rectangle_area(rctngl_base, rctngl_height));
        Console.WriteLine(tr.triangle_area(trngl_base, trngl_height));
    }
}
```

2. Solution

```
class Pet {
    public string kind;
    public int legs_number;

    public void start_running() {
        Console.WriteLine("Pet is running");
    }

    public void stop_running() {
        Console.WriteLine("Pet stopped");
    }
}

class Program {
    static void Main(string[] args) {
        Pet pet1 = new Pet();
        pet1.kind = "dog";
        pet1.legs_number = 4;

        Pet pet2 = new Pet();
        pet2.kind = "monkey";
        pet2.legs_number = 2;

        pet1.start_running();
        pet2.start_running();
        pet1.stop_running();
    }
}
```

3. Solution

```
class Pet {
    private string _kind;
    private int _legs_number;

    //Define the constructor
    public Pet(string kind, int legs_number) {
        this.Kind = kind;
        this.Legs_number = legs_number;
    }

    //Define public property Kind
    public string Kind {
        get {
            return this._kind;
        }

        set {
            if (value != "") {
                this._kind = value;
            }
        }
    }
}
```

```

        else {
            throw new Exception("Cannot be empty");
        }
    }
}

//Define public property Legs_number
public int Legs_number {
    get {
        return this._legs_number;
    }

    set {
        if (value >= 0) {
            this._legs_number = value;
        }
        else {
            throw new Exception("Cannot be negative");
        }
    }
}

public void start_running() {
    Console.WriteLine("Pet is running");
}

public void stop_running() {
    Console.WriteLine("Pet stopped");
}
}

class Program {
    static void Main(string[] args) {
        Pet pet1 = new Pet("dog", 4);

        pet1.start_running();
        pet1.stop_running();

        pet1.Kind = ""; //This will throw an error
        pet1.Legs_number = -1; //This will throw an error
    }
}

```

4. Solution

```

class Box {
    private double _width;
    private double _length;
    private double _height;

    //Define the constructor
    public Box(double w, double l, double h) {
        //Initialize fields
    }
}

```

```

        this._width = w;
        this._length = l;
        this._height = h;
    }

    public void display_volume() {
        Console.WriteLine("Volume: " + (this._width * this._length * this._height));
    }

    public void display_dimensions() {
        Console.WriteLine(this._width + " x " + this._length + " x " + this._height);
    }
}

class Program {
    const int BOXES = 3;

    static void Main(string[] args) {
        int i;
        double w, l, h;

        Box[] list_of_obj = new Box[BOXES]; //create an array

        for (i = 0; i <= BOXES - 1; i++) {
            Console.Write("Enter width: ");
            w = Double.Parse(Console.ReadLine());
            Console.Write("Enter length: ");
            l = Double.Parse(Console.ReadLine());
            Console.Write("Enter height: ");
            h = Double.Parse(Console.ReadLine());

            //Add each new object to the array
            list_of_obj[i] = new Box(w, l, h);
        }

        for (i = 0; i <= BOXES - 1; i++) {
            list_of_obj[i].display_dimensions();
            list_of_obj[i].display_volume();
        }
    }
}

```

5. Solution

```

class Box {
    private double _width;
    private double _length;
    private double _height;

    //Define the constructor
    public Box(double w, double l, double h) {
        //Initialize fields (using the corresponding properties)
        this.Width = w;
    }
}

```



```
this.Length = l;
this.Height = h;
}

//Define public property Width
public double Width {
    //Define the getter
    get {
        return this._width;
    }

    //Define the setter
    set {
        if (value > 0) {
            this._width = value;
        }
        else {
            throw new Exception("Cannot be negative or zero");
        }
    }
}

//Define public property Length
public double Length {
    //Define the getter
    get {
        return this._length;
    }

    //Define the setter
    set {
        if (value > 0) {
            this._length = value;
        }
        else {
            throw new Exception("Cannot be negative or zero");
        }
    }
}

//Define public property Height
public double Height {
    //Define the getter
    get {
        return this._height;
    }

    //Define the setter
    set {
        if (value > 0) {
            this._height = value;
        }
    }
}
```

```

        else {
            throw new Exception("Cannot be negative or zero");
        }
    }
}

public void display_volume() {
    Console.WriteLine("Volume: " + (this.Width * this.Length * this.Height));
}

public void display_dimensions() {
    Console.WriteLine(this.Width + " x " + this.Length + " x " + this.Height);
}
}

class Program {
    const int BOXES = 3;

    static void Main(string[] args) {
        int i;
        double w, l, h;

        Box[] list_of_obj = new Box[BOXES]; //Create an array

        for (i = 0; i <= BOXES - 1; i++) {
            Console.Write("Enter width: ");
            w = Double.Parse(Console.ReadLine());
            Console.Write("Enter length: ");
            l = Double.Parse(Console.ReadLine());
            Console.Write("Enter height: ");
            h = Double.Parse(Console.ReadLine());

            //Add each new object to the array
            list_of_obj[i] = new Box(w, l, h);
        }

        for (i = 0; i <= BOXES - 1; i++) {
            list_of_obj[i].display_dimensions();
            list_of_obj[i].display_volume();
        }
    }
}

```

6. Solution

```

class Cube {
    private double _edge;

    //Define the constructor
    public Cube(double edge) {
        this._edge = edge;
    }

    public void display_volume() {

```

```

        Console.WriteLine("Volume: " + Math.Pow(this._edge, 3));
    }

    public void display_one_surface() {
        Console.WriteLine("One surface: " + Math.Pow(this._edge, 2));
    }

    public void display_total_surface() {
        Console.WriteLine("Total surface: " + 6 * Math.Pow(this._edge, 2));
    }
}

class Program {
    static void Main(string[] args) {
        double edge;

        Console.Write("Enter edge length of a cube: ");
        edge = Double.Parse(Console.ReadLine());

        Cube cubel = new Cube(edge);

        cubel.display_volume();
        cubel.display_one_surface();
        cubel.display_total_surface();
    }
}

```

7. Solution

```

class Cube {
    private double _edge;

    //Define the constructor
    public Cube(double edge) {
        this.Edge = edge;
    }

    //Define public property Edge
    public double Edge {
        //Define the getter
        get {
            return this._edge;
        }

        //Define the setter
        set {
            if (value > 0) {
                this._edge = value;
            }
            else {
                throw new Exception("Cannot be negative or zero");
            }
        }
    }
}

```

```

    }

    public void display_volume() {
        Console.WriteLine("Volume: " + Math.Pow(this.Edge, 3));
    }

    public void display_one_surface() {
        Console.WriteLine("One surface: " + Math.Pow(this.Edge, 2));
    }

    public void display_total_surface() {
        Console.WriteLine("Total surface: " + 6 * Math.Pow(this.Edge, 2));
    }
}

class Program {
    static void Main(string[] args) {
        double edge;

        Console.Write("Enter edge length of a cube: ");
        edge = Double.Parse(Console.ReadLine());

        Cube cubel = new Cube(edge);

        cubel.display_volume();
        cubel.display_one_surface();
        cubel.display_total_surface();
    }
}

```

8. Solution

```

class Circle {
    private double _radius = -1;

    //Define public property Radius
    public double Radius {
        //Define the getter
        get {
            if (this._radius > 0) {
                return this._radius;
            }
            else {
                throw new Exception("Radius is not set");
            }
        }

        //Define the setter
        set {
            if (value > 0) {
                this._radius = value;
            }
            else {

```

```
        throw new Exception("Cannot be negative or zero");
    }
}

public double get_diameter() {
    return 2 * this.Radius;
}

public double get_area() {
    return 3.14 * Math.Pow(this.Radius, 2);
}

public double get_perimeter() {
    return 2 * 3.14 * this.Radius;
}
}

class Program {
    static void display_menu() {
        Console.WriteLine("1. Enter radius");
        Console.WriteLine("2. Display radius");
        Console.WriteLine("3. Display diameter");
        Console.WriteLine("4. Display area");
        Console.WriteLine("5. Display perimeter");
        Console.WriteLine("6. Exit");
    }

    static void Main(string[] args) {
        int choice;
        double radius;

        Circle circle1 = new Circle();

        while (true) {
            display_menu();
            Console.Write("Enter a choice: ");
            choice = Int32.Parse(Console.ReadLine());

            if (choice == 6) {
                Console.WriteLine("Bye");
                break;
            }
            else if (choice == 1) {
                Console.Write("Enter radius: ");
                radius = Double.Parse(Console.ReadLine());
                circle1.Radius = radius;
            }
            else if (choice == 2) {
                Console.WriteLine("Radius: " + circle1.Radius);
            }
            else if (choice == 3) {
                Console.WriteLine("Diameter: " + circle1.get_diameter());
            }
        }
    }
}
```

```

    }
    else if (choice == 4) {
        Console.WriteLine("Area: " + circle1.get_area());
    }
    else if (choice == 5) {
        Console.WriteLine("Perimeter: " + circle1.get_perimeter());
    }
}
}
}
}

```

9. Solution

```

class Info {
    private string _user_text;

    //Define public property User_text
    public string User_text {
        //Define the getter
        get {
            return this._user_text;
        }

        //Define the setter
        set {
            if (value != "") {
                this._user_text = value;
            }
            else {
                throw new Exception("Cannot be set to empty");
            }
        }
    }

    public int get_spaces_count() {
        int i, count = 0;
        string character;

        for (i = 0; i <= this.User_text.Length - 1; i++) {
            character = "" + this.User_text.ToLower()[i];
            if (character == " ") {
                count += 1;
            }
        }
        return count;
    }

    public int get_words_count() {
        return this.get_spaces_count() + 1;
    }

    public int get_vowels_count() {
        int i, count = 0;

```

```

    string character;

    for (i = 0; i <= this.User_text.Length - 1; i++) {
        character = "" + this.User_text[i];
        if ("aeiou".IndexOf(character) > -1) {
            count += 1;
        }
    }
    return count;
}

public int get_letters_count() {
    return this.User_text.Length - this.get_spaces_count();
}
}

class Program {
    static void Main(string[] args) {
        Info inf = new Info();

        Console.Write("Enter a text: ");
        inf.User_text = Console.ReadLine();

        Console.WriteLine("Text: " + inf.User_text);
        Console.WriteLine("Spaces: " + inf.get_spaces_count());
        Console.WriteLine("Words: " + inf.get_words_count());
        Console.WriteLine("Vowels: " + inf.get_vowels_count());
        Console.WriteLine("Total number of letters: " + inf.get_letters_count());
    }
}

```

10. Solution

```

class EncryptDecrypt {
    const string alphabet = " abcdefghijklmnopqrstuvwxyz"; //space is a valid character!
    private int _encr_decr_key = -1;

    //Define the property Encr_decr_key
    public int Encr_decr_key {
        //Define the getter
        get {
            if (this._encr_decr_key != -1) {
                return this._encr_decr_key;
            }
            else {
                throw new Exception("Key is not set");
            }
        }
    }

    //Define the setter
    set {
        if (value >= 1 && value <= 26) {
            this._encr_decr_key = value;
        }
    }
}

```

```

    }
    else {
        throw new Exception("Must be between 1 and 26");
    }
}
}

public string encrypt(string message) {
    string character, return_value = "";
    char new_letter;
    int i, index, new_index;

    for (i = 0; i <= message.Length - 1; i++) {
        character = "" + message[i];
        index = alphabet.IndexOf(character);
        new_index = index + this.Encr_decr_key;
        if (new_index >= 27) {
            new_index -= 27;
        }
        new_letter = alphabet[new_index];
        return_value += new_letter;
    }
    return return_value;
}

public string decrypt(string enc_message) {
    string character, return_value = "";
    char new_letter;
    int i, index, new_index;

    for (i = 0; i <= enc_message.Length - 1; i++) {
        character = "" + enc_message[i];
        index = alphabet.IndexOf(character);
        new_index = index - this.Encr_decr_key;
        if (new_index < 0) {
            new_index += 27;
        }
        new_letter = alphabet[new_index];
        return_value += new_letter;
    }
    return return_value;
}
}

class Program {
    static void display_menu() {
        Console.WriteLine("1. Enter encryption/decryption key");
        Console.WriteLine("2. Encrypt a message");
        Console.WriteLine("3. Decrypt a message");
        Console.WriteLine("4. Exit");
    }
}

```



```
static void Main(string[] args) {
    string text;
    int choice;

    EncryptDecrypt ed = new EncryptDecrypt();

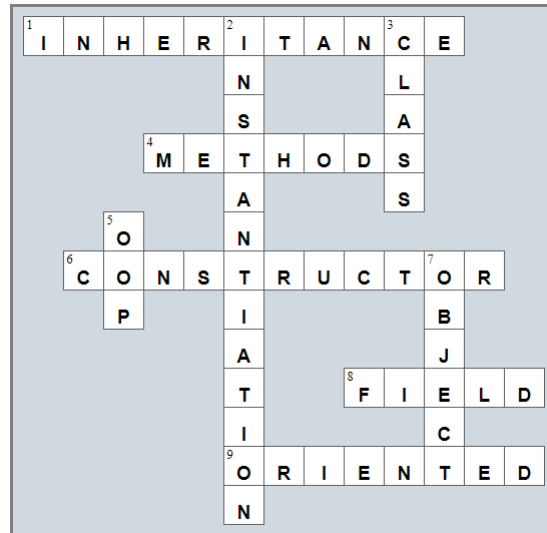
    display_menu();
    Console.Write("Enter a choice: ");
    choice = Int32.Parse(Console.ReadLine());
    while (choice != 4) {
        if (choice == 1) {
            Console.Write("Enter encryption/decryption key: ");
            ed.Encr_decr_key = Int32.Parse(Console.ReadLine());
        }
        else if (choice == 2) {
            Console.Write("Enter message to encrypt: ");
            text = Console.ReadLine();
            Console.WriteLine("Encrypted message: " + ed.encrypt(text));
        }
        else if (choice == 3) {
            Console.Write("Enter message to decrypt: ");
            text = Console.ReadLine();
            Console.WriteLine("Decrypted message: " + ed.decrypt(text));
        }

        display_menu();
        Console.Write("Enter a choice: ");
        choice = Int32.Parse(Console.ReadLine());
    }
}
```

Review in "Object Oriented Programming"

Review Crossword Puzzle

1.



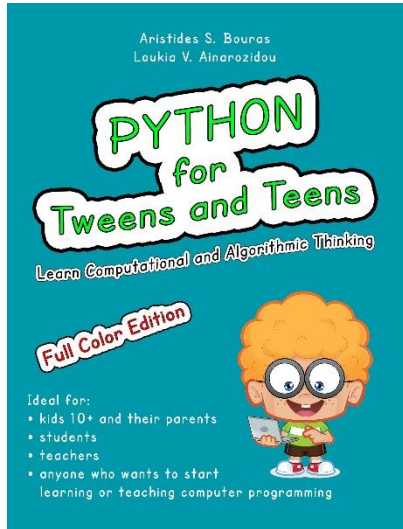
Some Final Words from the Author

I hope you really enjoyed reading this book. I made every possible effort to make it comprehensible even by people that probably have no previous experience in programming.

So if you liked this book, please visit the web store where you bought it and show me your gratitude by writing a good review and giving me as many stars as possible. By doing this, you will encourage me to continue writing and of course you'll help other readers to reach me.

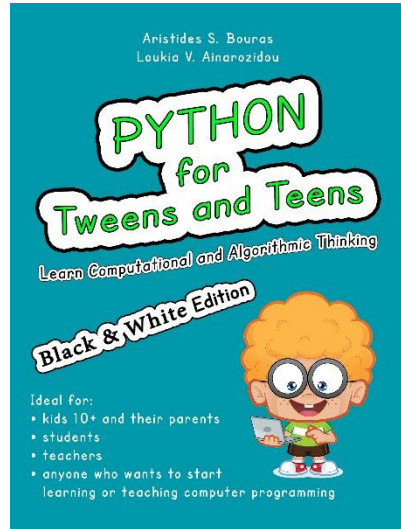
And remember: Learning is a process within an endless loop. It begins at birth and continues throughout your lifetime!

Some of my Books



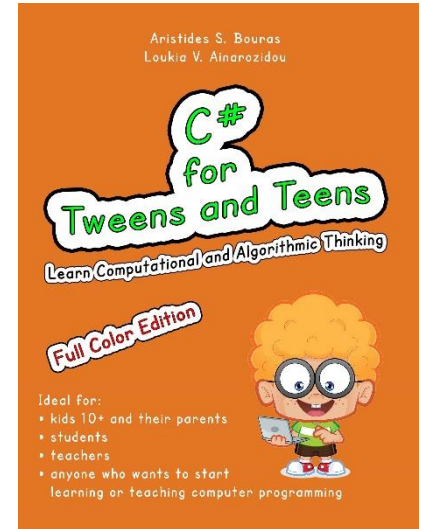
ISBN-10: 1543127940

ISBN-13: 978-1543127942



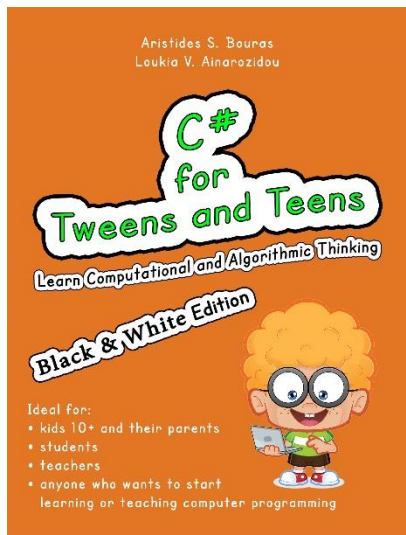
ISBN-10: 1546611215

ISBN-13: 978-1546611219



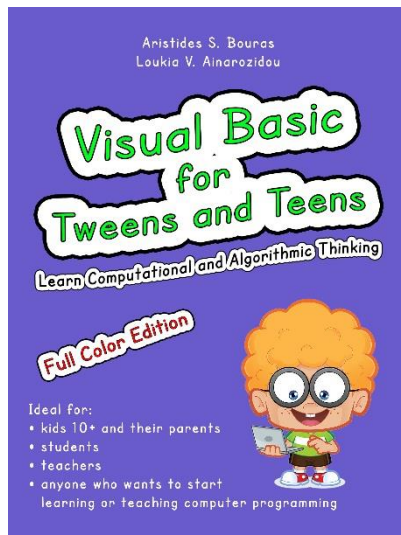
ISBN-10: 1973727684

ISBN-13: 978-1973727682



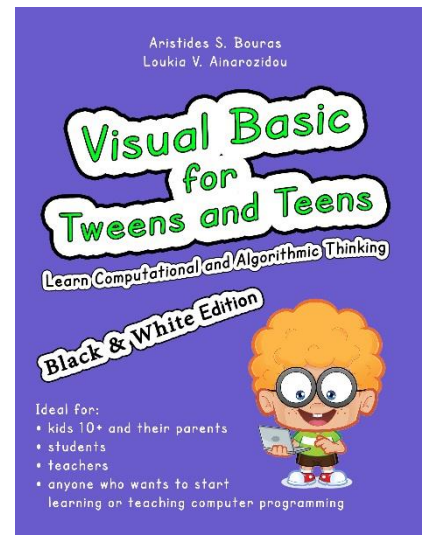
ISBN-10: 1973727765

ISBN-13: 978-1973727767



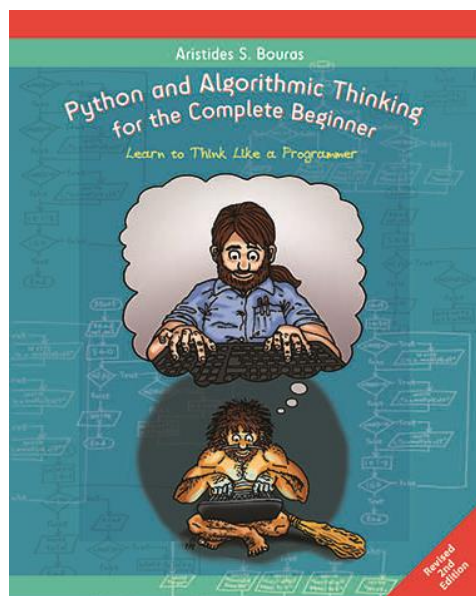
ISBN-10: 1982083670

ISBN-13: 978-1982083670



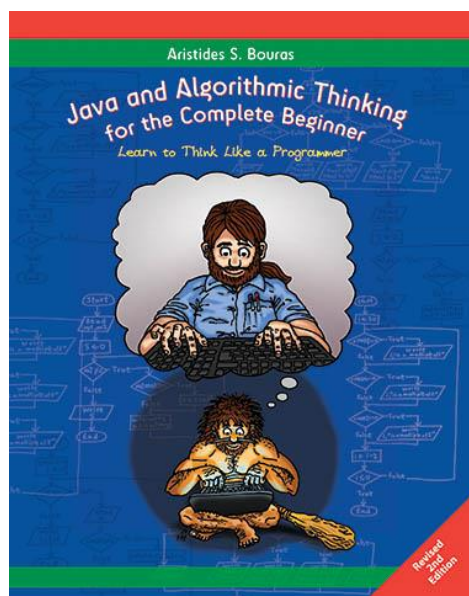
ISBN-10: 1982083697

ISBN-13: 978-1982083694



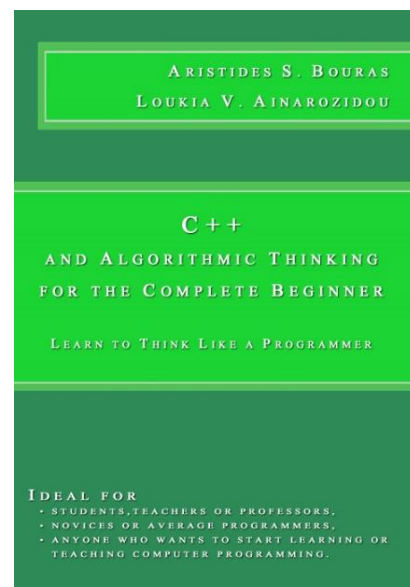
ISBN-10: 1099184878

ISBN-13: 978-1099184871



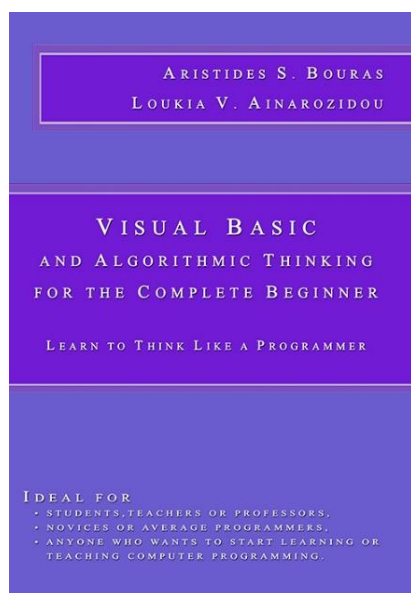
ISBN-10: 1698811233

ISBN-13: 978-1698811239



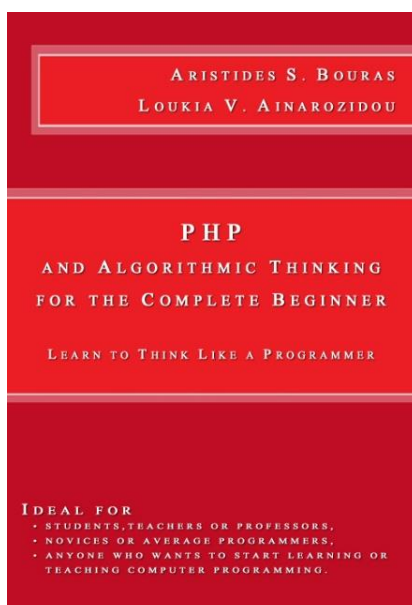
ISBN-10: 1508577552

ISBN-13: 978-1508577553



ISBN-10: 1511798963

ISBN-13: 978-1511798969



ISBN-10: 1503015912

ISBN-13: 978-1503015913