C#
AND ALGORITHMIC THINKING
FOR THE COMPLETE BEGINNER

Compact Edition

The Answers

Aristides S. Bouras
Loukia V. Ainarozidou

Copyright ©

Copyright © by Aristides S. Bouras and Loukia V. Ainarozidou

https://www.bouraspage.com

RCode: 181204

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 – Compact Edition”. Every effort has been taken to make this book compatible with all previous releases of C#, and it is almost certain to be compatible with any future releases of C#.

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

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

https://www.bouraspage.com

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

Chapter 1

1.7 Answers of Review Questions: True/False

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

1.8 Answers of Review Questions: Multiple Choice

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

Chapter 4

4.16 Answers of Review Questions: True/False

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

4.17 Answers of Review Questions: Multiple Choice

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

Chapter 5

5.9 Answers of Review Questions: True/False

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

5.10 Answers of Review Questions: Multiple Choice

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

5.11 Answers of Review Exercises

1. 1 – c, 2 – d, 3 – a, 4 – b
2. 1 – d, 2 – c, 3 – b, 4 – a

Chapter 6

6.4 Answers of Review Questions: True/False

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

6.5 Answers of Review Questions: Multiple Choice

1. a
2. b
3. b

Chapter 7

7.7 Answers of Review Questions: True/False

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

7.8 Answers of Review Questions: Multiple Choice

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

7.9 Answers of Review Exercises

1. ii, iv, v, ix
2. i. String, ii. Boolean, iii. String, iv. String, v. Real, vi. Integer
3. i. d, ii. f, iii. c, iv. e
4. i. 26, ii. 28
5. i. 5, ii. 6
6. i. 1, ii. 0, iii. 1, iv. 1, v. 0, vi. 1
7. i. 2 * 3, ii. 4
8. i. 2, ii. 0, iii. 1, iv. 0, v. Division by zero error, vi. 0
9. i. 2, ii. 10.5
10. My name is George Malkovich
11. i. (-3), ii. 1
12. California, California

Chapter 8

8.2 Answers of Review Questions: True/False

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

8.3 Answers of Review Exercises

1. Solution

For the input value of 3

Step	Statement	a	b	c	d
1	`a = Int32.Parse(Console.ReadLine())`	**3**	?	?	?
2	`a = (a + 1) * (a + 1) + 6 / 3 * 2 + 20`	**40**	?	?	?
3	`b = a % 13`	40	**1**	?	?
4	`c = b % 7`	40	1	**1**	?
5	`d = a * b * c`	40	1	1	**40**
6	`Console.Write(a + ", " + b + ", " + c + ", " + d)`	40, 1, 1, 40 is displayed			

For the input value of 4

Step	Statement	a	b	c	d
1	`a = Int32.Parse(Console.ReadLine())`	**4**	?	?	?
2	`a = (a + 1) * (a + 1) + 6 / 3 * 2 + 20`	**49**	?	?	?
3	`b = a % 13`	49	**10**	?	?
4	`c = b % 7`	49	10	**3**	?
5	`d = a * b * c`	49	10	3	**1470**
6	`Console.Write(a + ", " + b + ", " + c + ", " + d)`	49, 10, 3, 1470 is displayed			

For the input value of 1

Step	Statement	a	b	c	d
1	`a = Int32.Parse(Console.ReadLine())`	**1**	?	?	?
2	`a = (a + 1) * (a + 1) + 6 / 3 * 2 + 20`	**28**	?	?	?
3	`b = a % 13`	28	**2**	?	?
4	`c = b % 7`	28	2	**2**	?
5	`d = a * b * c`	28	2	2	**112**
6	`Console.Write(a + ", " + b + ", " + c + ", " + d)`	28, 2, 2, 112 is displayed			

2. *Solution*

For the input values of 8, 4

Step	Statement	a	b	c	d	e
1	`a = Int32.Parse(Console.ReadLine())`	**8**	?	?	?	?
2	`b = Int32.Parse(Console.ReadLine())`	8	**4**	?	?	?
3	`c = a + b`	8	4	**12**	?	?
4	`d = 1 + a / b * c + 2`	8	4	12	**27**	?
5	`e = c + d`	8	4	12	27	**39**
6	`c += d + e`	8	4	**78**	27	39
7	`e--`	8	4	78	27	**38**
8	`d -= c + c % d`	8	4	78	**-75**	38
9	`Console.Write(c + ", " + d + ", " + e)`	78, -75, 38 is displayed				

For the input values of 4, 4

Step	Statement	a	b	c	d	e
1	`a = Int32.Parse(Console.ReadLine())`	**4**	?	?	?	?
2	`b = Int32.Parse(Console.ReadLine())`	4	**4**	?	?	?
3	`c = a + b`	4	4	**8**	?	?
4	`d = 1 + a / b * c + 2`	4	4	8	**11**	?
5	`e = c + d`	4	4	8	11	**19**
6	`c += d + e`	4	4	**38**	11	19
7	`e--`	4	4	38	11	**18**
8	`d -= c + c % d`	4	4	38	**-32**	18
9	`Console.Write(c + ", " + d + ", " + e)`	38, -32, 18 is displayed				

Chapter 9

9.5 Answers of 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	`a = Int32.Parse(Console.ReadLine())`	**5**	?	?	?	?
2	`b = Int32.Parse(Console.ReadLine())`	5	**5**	?	?	?
3	`c = a + b`	5	5	**10**	?	?
4	`d = 5 + a / b * c + 2`	5	5	10	**17**	?
5	`e = c - d`	5	5	10	17	**-7**
6	`c += d + c`	5	5	**37**	17	-7
7	`e--`	5	5	37	17	**-8**
8	`d += e + c % b`	5	5	37	**11**	-8
9	`Console.Write(c + ", " + d + ", " + e)`	37, 11, -8 is displayed				

For the input values of 4, 2

Step	Statement	a	b	c	d	e
1	`a = Int32.Parse(Console.ReadLine())`	**4**	?	?	?	?
2	`b = Int32.Parse(Console.ReadLine())`	4	**2**	?	?	?
3	`c = a + b`	4	2	**12**	?	?
4	`d = 5 + a / b * c + 2`	4	2	12	**19**	?
5	`e = c - d`	4	2	12	19	**-13**
6	`c += d + c`	4	2	**31**	19	-13
7	`e--`	4	2	31	19	**-14**
8	`d += e + c % b`	4	2	31	**6**	-14
9	`Console.Write(c + ", " + d + ", " + e)`	31, 6, -14 is displayed				

3. Solution

For the input value of 5

Step	Statement	a	b	c
1	`b = Int32.Parse(Console.ReadLine())`	?	**5**	?
2	`c = 5`	?	5	**5**

3	`c = c * b`	?	5	**25**
4	`a = 3 * c % 10`	**5**	5	25
5	`Console.Write(a)`	Value 5 is displayed		

For the input value of 4

Step	Statement	a	b	c
1	`b = Int32.Parse(Console.ReadLine())`	?	**4**	?
2	`c = 5`	?	4	**5**
3	`c = c * b`	?	4	**20**
4	`a = 3 * c % 10`	**0**	4	20
5	`Console.Write(a)`	Value 0 is displayed		

For the input value of 15

Step	Statement	a	b	c
1	`b = Int32.Parse(Console.ReadLine())`	?	15	?
2	`c = 5`	?	15	**5**
3	`c = c * b`	?	15	**75**
4	`a = 3 * c % 10`	**5**	15	75
5	`Console.Write(a)`	Value 5 is displayed		

Chapter 10

10.2 Answers of Review Exercises

1. Solution

```
static void Main() {
  double base, height, area;

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

  area = 0.5 * base * height;

  Console.Write(area);
  Console.ReadKey();
}
```

2. Solution

```
static void Main() {
  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.Write(angle3);
  Console.ReadKey();
}
```

3. Solution

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

  Console.Write("Enter 1st grade: ");
  g1 = Int32.Parse(Console.ReadLine());
  Console.Write("Enter 2nd grade: ");
  g2 = Int32.Parse(Console.ReadLine());
  Console.Write("Enter 3rd grade: ");
  g3 = Int32.Parse(Console.ReadLine());
  Console.Write("Enter 4th grade: ");
  g4 = Int32.Parse(Console.ReadLine());

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

  Console.Write(average);
  Console.ReadKey();
}
```

4. Solution

```
const int PI = 3.14159;

static void Main() {
  double r, perimeter;

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

  perimeter = 2 * PI * r;

  Console.Write(perimeter);
  Console.ReadKey();
}
```

5. Solution

```
static void Main() {
  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.Write(total);
  Console.ReadKey();
}
```

6. Solution

```
static void Main() {
  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.Write(s);
  Console.ReadKey();
}
```

7. Solution

```
static void Main() {
  double f, c;

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

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

  Console.Write(c);
  Console.ReadKey();
}
```

8. Solution

```
static void Main() {
  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.Write(bmi);
  Console.ReadKey();
}
```

9. Solution

```
static void Main() {
  double s_total, g_rate, tip, total, ;

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

  tip = s_total * g_rate / 100;

  total = s_total + tip;

  Console.WriteLine("Tip is " + tip);
  Console.Write(" and Total is " + total);
  Console.ReadKey();
}
```

10. Solution

```
const double VAT = 0.20;
```

```
static void Main() {
  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.Write(avg);
  Console.ReadKey();
}
```

11. Solution

```
const int VAT = 0.20;

static void Main() {
  double atax_price, btax_price;

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

  btax_price = atax_price / (1 + VAT);

  Console.Write(btax_price);
  Console.ReadKey();
}
```

12. Solution

```
static void Main() {
  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.Write(f_price + " " + saved);
  Console.ReadKey();
}
```

13. Solution

```
const int VAT = 0.20;

static void Main() {
  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.Write(kWh_consumed + " " + cost);
  Console.ReadKey();
}
```

14. Solution

```
static void Main() {
  int day, month, days_passed, days_left;

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

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

  Console.Write(days_left);
  Console.ReadKey();
}
```

Chapter 11

11.3 Answers of Review Questions: True/False

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

11.4 Answers of Review Questions: Multiple Choice

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

11.5 Answers of Review Exercises

1. Solution

For the input value of 9

Step	Statement	a	b	c
1	`a = Int32.Parse(Console.ReadLine())`	**9**	?	?
2	`a += 6 / Math.Sqrt(a) * 2 + 20`	**33**	?	**?**
3	`b = (int)Math.Round(a) % 4`	33	**1**	**?**
4	`c = b % 3`	33	1	**1**
5	`Console.Write(a + ", " + b + ", " + c)`	33, 1, 1 is displayed		

For the input value of 4

Step	Statement	a	b	c
1	`a = Int32.Parse(Console.ReadLine())`	**4**	?	?
2	`a += 6 / Math.Sqrt(a) * 2 + 20`	**30**	?	**?**
3	`b = (int)Math.Round(a) % 4`	30	**2**	**?**
4	`c = b % 3`	30	2	**2**
5	`Console.Write(a + ", " + b + ", " + c)`	30, 2, 2 is displayed		

2. Solution

For the input value of -2

Step	Statement	a	b	c
1	`a = Int32.Parse(Console.ReadLine())`	**-2**	?	?
2	`b = (int)Math.Abs(a) % 4 + (int)Math.Pow(a , 4)`	-2	**18**	**?**
3	`c = b % 5`	-2	18	**3**

4	`Console.Write(b + ", " + c)`	18, 3 is displayed		

For the input value of -3

Step	Statement	a	b	c
1	`a = Int32.Parse(Console.ReadLine())`	**-3**	?	?
2	`b = (int)Math.Abs(a) % 4 + (int)Math.Pow(a , 4)`	-3	**84**	**?**
3	`c = b % 5`	-3	84	**4**
4	`Console.Write(b + ", " + c)`	84, 4 is displayed		

3. Solution

```
static void Main() {
  double degrees, radians;

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

  degrees = radians * 180 / Math.PI;

  Console.Write(degrees);
  Console.ReadKey();
}
```

4. Solution

```
static void Main() {
  double a, b, hypotenuse;

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

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

  Console.Write(hypotenuse);
  Console.ReadKey();
}
```

5. Solution

```
static void Main() {
  double adjacent, opposite, th;

  Console.Write("Enter angle (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.Write(opposite);
  Console.ReadKey();
}
```

Chapter 12

12.2 Answers of Review Exercises

1. *Solution*

i. a, e, g, h

ii. c, f

2. *Solution*

i. `y = Math.Pow(x + 3, 5 * w) / (7 * (x - 4))`

ii. `y = Math.Pow(3 * Math.Pow(x, 2) - Math.Pow(x, 3) / 4, 1 / 5.0)`

iii. `y = Math.Sqrt(Math.Pow(x, 4) - 2 * Math.Pow(x, 3) - 7 * x * x + x) / Math.Pow(4 * (7 * Math.Pow(x, 4) - 3 / 4.0 * Math.Pow(x, 3)) * (7 * x * x + x), 1/3.0)`

iv. `y = x / (x - 3 * (x - 1)) + x * Math.Pow(x - 1, 1 / 5.0) / ((Math.Pow(x, 3) - 2) * Math.Pow(x - 1, 3))`

v. `y = Math.Pow(Math.Sin(Math.PI / 3) - Math.Cos(Math.PI / 2 * w) , 2)`

vi. `y = Math.Pow(Math.Sin(Math.PI / 2 * x) + Math.Cos(3 * Math.PI / 2 * w), 3) / Math.Pow(Math.Tan(2 * Math.PI / 3 * w) - Math.Sin(Math.PI / 2 * x), 0.5)`

3. *Solution*

```
static void Main() {
  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.Write(y);
  Console.ReadKey();
}
```

4. *Solution*

```
static void Main() {
  double x, y;

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

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

  Console.Write(y);
  Console.ReadKey();
}
```

5. Solution

```
static void Main() {
  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.Write(y);
  Console.ReadKey();
}
```

6. Solution

```
static void Main() {
  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.Write(y);
  Console.ReadKey();
}
```

7. Solution

```
static void Main() {
  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.Write(y);
  Console.ReadKey();
}
```

8. Solution

```
static void Main() {
  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.Write(area);
  Console.ReadKey();
}
```

Chapter 13

13.2 Answers of Review Exercises

1. Solution

```
static void Main() {
  int last_digit, n, result;

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

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

  Console.Write(result);
  Console.ReadKey();
}
```

2. Solution

```
static void Main() {
  int digit1, digit2, digit3, digit4, digit5, number, r, reversed;

  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 = digit5 * 10000 + digit4 * 1000 + digit3 * 100 + digit2 * 10 + digit1;
  Console.Write(reversed);
  Console.ReadKey();
}
```

3. Solution

```
static void Main() {
  int n, result;

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

  result = n % 2;

  Console.Write(result);
  Console.ReadKey();
}
```

4. Solution

```
static void Main() {
  int n, result;

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

  result = 1 - n % 2;

  Console.Write(result);
  Console.ReadKey();
}
```

5. Solution

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

  Console.Write("Enter a period of 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 + " weeks " + days + " days " + hours + " hours ");
  Console.Write(minutes + " minutes and " + seconds + " seconds");
  Console.ReadKey();
}
```

6. Solution

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

  Console.Write("Enter amount 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 + " notes of $20 " + usd10 + " notes of $10 ");
  Console.Write(usd5 + " notes of $5 and " + usd1 + " notes of $1");
  Console.ReadKey();
}
```

7. Solution

```
static void Main() {
  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 + " miles " + yards + " yards ");
  Console.Write(feet + " feet and " + inches + " inches");
  Console.ReadKey();
}
```

Chapter 14

14.5 Answers of Review Questions: True/False

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

14.6 Answers of Review Questions: Multiple Choice

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

14.7 Answers of Review Exercises

1. *Solution*

```
static void Main() {
  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.Write(first_name + " " + last_name);
  Console.ReadKey();
}
```

2. *Solution*

```
static void Main() {
  string alphabet;
  Random rnd = new Random();

  alphabet = "abcdefghijklmnopqrstuvwxyz";

  Console.Write(alphabet[rnd.Next(0, 26)].ToString().ToUpper());
  Console.Write(alphabet[rnd.Next(0, 26)]);
  Console.Write(alphabet[rnd.Next(0, 26)]);
  Console.Write(alphabet[rnd.Next(0, 26)]);
  Console.Write(alphabet[rnd.Next(0, 26)]);

  Console.ReadKey();
}
```

3. Solution

```
static void Main() {
  string name;
  Random rnd = new Random();

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

  Console.Write(name[rnd.Next(0, name.Length)]);
  Console.Write(name[rnd.Next(0, name.Length)]);
  Console.Write(name[rnd.Next(0, name.Length)]);
  Console.Write(rnd.Next(1000, 10000));

  Console.ReadKey();
}
```

Chapter 15

15.8 Answers of Review Questions: True/False

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

15.9 Answers of Review Questions: Multiple Choice

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

15.10 Answers of 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
`Math.Pow(x + y, 3)`	8
`(x + y) / (Math.Pow(x, 2) - 14)`	1
`x - 1 == y + 5`	`true`
`x > 2 && y == 1`	`false`
`x == 1 || y == -2 && !(flag == false)`	`true`
`!(x >= 3) && (x % 2 > 1)`	`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 Answers of Review Questions: True/False

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

16.3 Answers of Review Questions: Multiple Choice

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

16.4 Answers of Review Exercises

1. Solution

The corrections/additions are in red

```
static void Main() {
  double x, y;

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

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

  Console.Write(x + " " + y);
  Console.ReadKey();
}
```

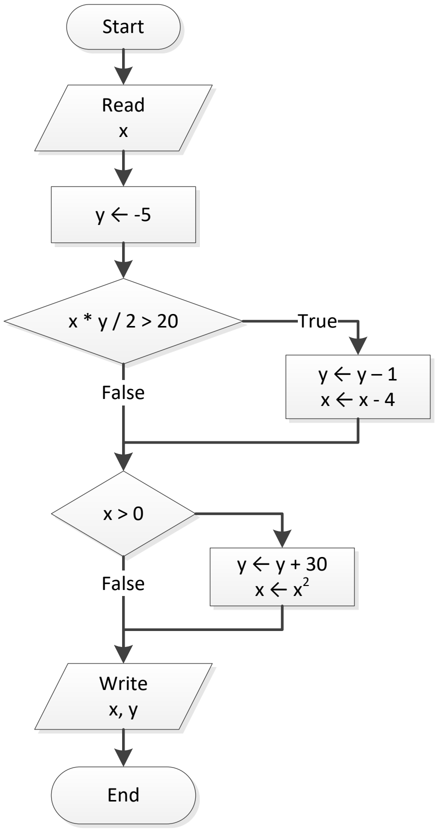
2. Solution

For the input value of 10

Step	Statement	x	y
1	`x = Double.Parse(Console.ReadLine())`	**10**	?
2	`y = - 5`	10	**-5**
3	`if (x * y / 2 > 20)`	False	
4	`if (x > 0)`	True	
5	`y += 30`	10	**25**
6	`x = Math.Pow(x, 2)`	**100**	25
7	`Console.Write(x + ", " + y)`	100, 25 is displayed	

For the input value of -10

Step	Statement	x	y
1	x = Double.Parse(Console.ReadLine())	**-10**	?
2	y = - 5	-10	**-5**
3	if (x * y / 2 > 20)	True	
4	y--	-10	**-6**
5	x -= 4	**-14**	-6
6	if (x > 0)	False	
7	Console.Write(x + ", " + y)	-14, -6 is displayed	



3. *Solution*

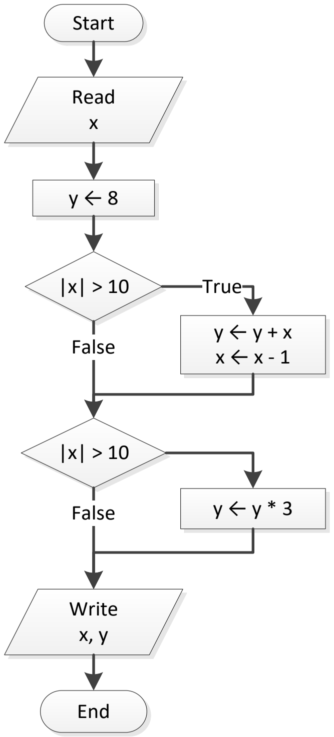
For the input value of -11

Step	Statement	x	y
1	x = Int32.Parse(Console.ReadLine())	**-11**	?
2	y = 8	-11	**8**
3	if (Math.Abs(x) > 10)	True	
4	y += x	-11	**-3**
5	x--	**-12**	-3

6	`if (Math.Abs(x) > 10)`	True	
7	`y *= 3`	-12	**-9**
8	`Console.Write(x + ", " + y)`	-12, -9 is displayed	

For the input value of 11

Step	Statement	x	y
1	`x = Int32.Parse(Console.ReadLine())`	**11**	?
2	`y = 8`	11	**8**
3	`if (Math.Abs(x) > 10)`	True	
4	`y += x`	11	**19**
5	`x--`	**10**	19
6	`if (Math.Abs(x) > 10)`	False	
7	`Console.Write(x + ", " + y)`	10, 19 is displayed	



4. Solution

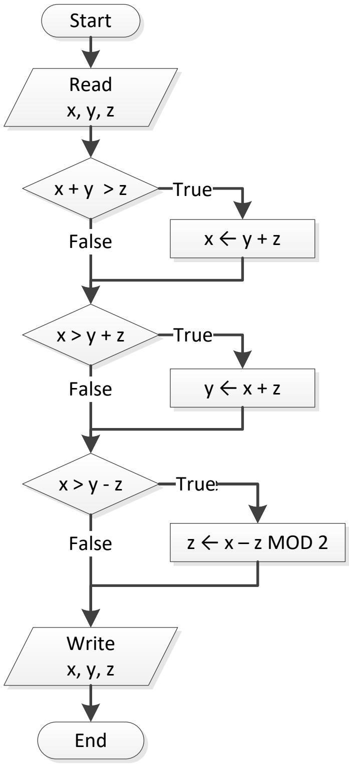
For input values of 1, 2 and 3

Step	Statement	x	y	z
1	`x = Int32.Parse(Console.ReadLine())`	**1**	?	?
2	`y = Int32.Parse(Console.ReadLine())`	1	**2**	?
3	`z = Int32.Parse(Console.ReadLine())`	1	**2**	**3**
4	`if (x + y > z)`	False		

5	`if (x > y + z)`		False	
6	`if (x > y - z)`		True	
7	`z = x - z % 2`	1	2	**0**
8	`Console.Write(x + ", " + y + ", " + z)`	1, 2, 0 is displayed		

For input values of 4, 2 and 1

Step	Statement	x	y	z
1	`x = Int32.Parse(Console.ReadLine())`	**4**	?	?
2	`y = Int32.Parse(Console.ReadLine())`	4	**2**	?
3	`z = Int32.Parse(Console.ReadLine())`	4	**2**	**1**
4	`if (x + y > z)`		True	
5	`x = y + z`	**3**	2	1
6	`if (x > y + z)`		False	
7	`if (x > y - z)`		True	
8	`z = x - z % 2`	3	2	**2**
9	`Console.Write(x + ", " + y + ", " + z)`	3, 2, 2 is displayed		



5. Solution

```
static void Main() {
  double x;

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

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

6. Solution

```
static void Main() {
  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");
  }
  Console.ReadKey();
}
```

7. Solution

```
static void Main() {
  string x;
  double x_dbl;

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

  if (Double.TryParse(x, out x_dbl) == true) {
    Console.WriteLine("Numeric");
  }

  Console.ReadKey();
}
```

8. Solution

```
static void Main() {
  string str;

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

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

9. Solution

```
static void Main() {
  string str;

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

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

10. Solution

```
static void Main() {
  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!");
  }
  Console.ReadKey();
}
```

11. Solution

```
static void Main() {
  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.Write(a + ", " + b);
  Console.ReadKey();
}
```

12. Solution

```
static void Main() {
  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");
  }
  Console.ReadKey();
}
```

Chapter 17

17.2 Answers of Review Questions: True/False

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

17.3 Answers of Review Questions: Multiple Choice

1. b
2. c
3. c

17.4 Answers of Review Exercises

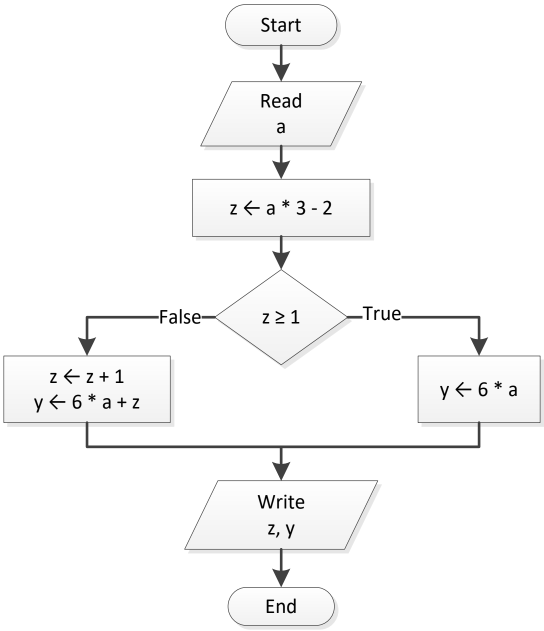
1. Solution

For input value of 3

Step	Statement	a	y	z
1	`a = Double.Parse(Console.ReadLine())`	**3**	?	?
2	`z = a * 3 - 2`	3	?	**7**
3	`if (z >= 1)`	True		
4	`y = 6 * a`	3	**18**	7
5	`Console.Write(z + ", " + y)`	7, 18 is displayed		

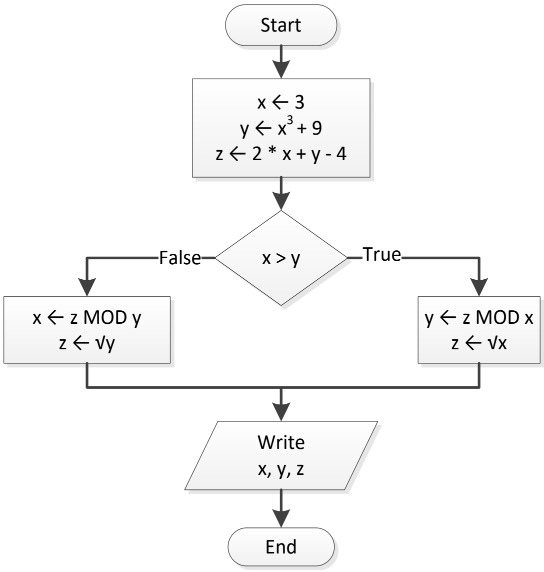
For input value of 0.5

Step	Statement	a	y	z
1	`a = Double.Parse(Console.ReadLine())`	**0.5**	?	?
2	`z = a * 3 - 2`	0.5	?	**-0.5**
3	`if (z >= 1)`	False		
4	`z++`	0.5	?	**0.5**
5	`y = 6 * a + z`	0.5	**3.5**	0.5
6	`Console.Write(z + ", " + y)`	0.5, 3.5 is displayed		



2. *Solution*

Step	Statement	x	y	z
1	`x = 3`	**3**	?	?
2	`y = Math.Pow(x, 3) + 9`	3	**36**	?
3	`z = 2 * x + y - 4`	3	36	**38**
4	`if (x > y)`	False		
5	`x = z % y`	**2**	36	38
6	`z = Math.Sqrt(y)`	2	36	**6**
7	`Console.Write(x + ", " + y + ", " + z)`	2, 36, 6 is displayed		



3. Solution

```
static void Main() {
  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.Write(y);
  Console.ReadKey();
}
```

For input value of 10

Step	Statement	x	y	w	z
1	`x = Double.Parse(Console.ReadLine())`	**10**	?	?	?
2	`w = x * 3 - 15`	10	?	**15**	?
3	`z = (w + 7) * (x + 4) - 10`	10	?	15	**298**
4	`if (w > x && z > x)`	True			
5	`x++`	**11**	?	15	298
6	`y = x / 2 + 4`	11	**9.5**	15	298
7	`Console.Write(y)`	9.5 is displayed			

For input value of 2

Step	Statement	x	y	w	z
1	`x = Double.Parse(Console.ReadLine())`	**2**	?	?	?
2	`w = x * 3 - 15`	2	?	**-9**	?
3	`z = (w + 7) * (x + 4) - 10`	2	?	-9	**-22**
4	`if (w > x && z > x)`	False			
5	`y = x / 4 + 2`	2	**2.5**	-9	-22
6	`Console.Write(y)`	2.5 is displayed			

4. Solution

```
static void Main() {
  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");
  }
  Console.ReadKey();
}
```

5. Solution

```
static void Main() {
  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");
  }
  Console.ReadKey();
}
```

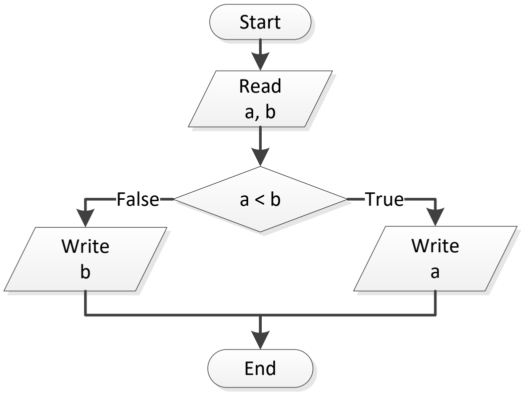
6. Solution

```
static void Main() {
  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.Write("The structure is: " + x + " = " + (int)(x / 4) + " x 4 + " + y);
  Console.ReadKey();
}
```

7. Solution

```
static void Main() {
  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");
  }
  Console.ReadKey();
}
```

8. Solution



```
static void Main() {
  double a, b;

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

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

9. Solution

```
static void Main() {
  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");
  }
  Console.ReadKey();
}
```

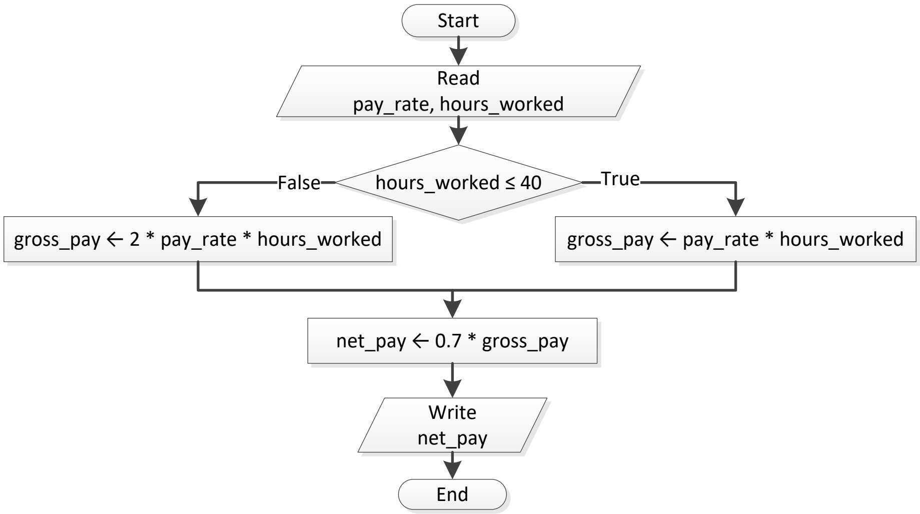
10. Solution

```
static void Main() {
  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");
  }
  Console.ReadKey();
}
```

11. Solution

```
static void Main() {
  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("Disqualified");
  }
  else {
    Console.WriteLine("Qualified");
  }
  Console.ReadKey();
}
```

12. Solution


Start
Read
pay_rate, hours_worked
hours_worked ≤ 40
False
True
gross_pay ← 2 * pay_rate * hours_worked
gross_pay ← pay_rate * hours_worked
net_pay ← 0.7 * gross_pay
Write
net_pay
End

```
static void Main() {
  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 = 2 * pay_rate * hours_worked;
  }

  net_pay = 0.7 * gross_pay;
  Console.Write("Net Pay: " + net_pay);
  Console.ReadKey();
}
```

13. Solution

```
static void Main() {
  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");
  }
  Console.ReadKey();
}
```

14. Solution

```
static void Main() {
  double a1, a2, s1, s2, t;

  Console.Write("Enter the time the two cars traveled: ");
  t = Double.Parse(Console.ReadLine());
  Console.Write("Enter the accelaration for car A: ");
  a1 = Double.Parse(Console.ReadLine());
  Console.Write("Enter the accelaration for car B: ");
  a2 = Double.Parse(Console.ReadLine());

  s1 = 0.5 * a1 * t;
  s2 = 0.5 * a2 * t;

  if (s1 > s2) {
    Console.WriteLine("Car A is first");
  }
  else {
    Console.WriteLine("Car B is first");
  }
  Console.ReadKey();
}
```

Chapter 18

18.2 Answers of Review Questions: True/False

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

18.3 Answers of 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.Write(b)`	1 is displayed	

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.Write(b)`	3 is displayed	

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.Write(b)`	4 is displayed	

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.Write(b)`	4 is displayed	

2. *Solution*

For input value of 5

Step	Statement	amount	discount	payment
1	`amount = Double.Parse(Console.ReadLine())`	**5**	?	?
2	`discount = 0`	5	**0**	?
3	`if (amount < 20)`	True		
4	`discount = 0`	5	**0**	?
5	`payment = amount - amount * discount / 100`	5	0	**5**
6	`Console.Write(discount + ", " + payment)`	0, 5 is displayed.		

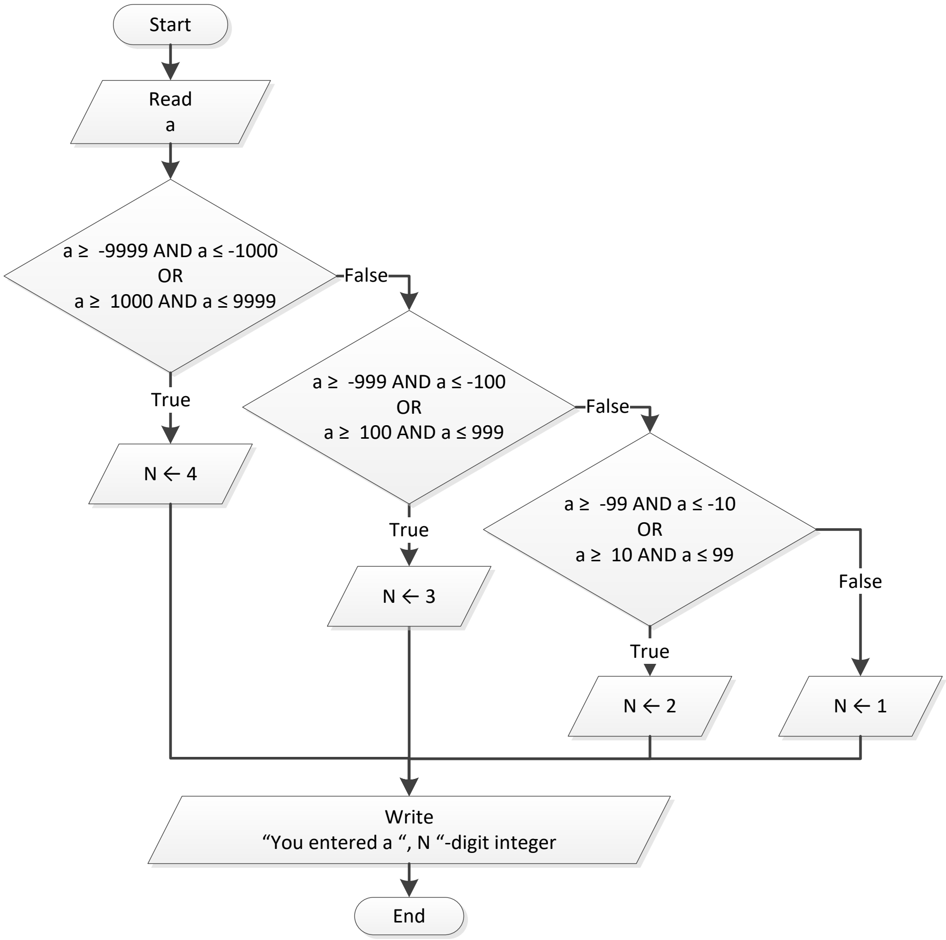
For input value of 150

Step	Statement	amount	discount	payment
1	`amount = Double.Parse(Console.ReadLine())`	**150**	?	?
2	`discount = 0`	150	**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	**15**	?
8	`payment = amount - amount * discount / 100`	150	15	**5**
9	`Console.Write(discount + ", " + payment)`	15, 127.5 is displayed.		

For input value of -1

Step	Statement	amount	discount	payment
1	`amount = Double.Parse(Console.ReadLine())`	**-1**	?	?
2	`discount = 0`	-1	**0**	?
3	`if (amount < 20)`	True		
4	`discount = 0`	-1	**0**	?
5	`payment = amount - amount * discount / 100`	-1	0	**-1**
6	`Console.Write(discount + ", " + payment)`	0, -1 is displayed.		

3. Solution



```
static void Main() {
  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.Write("You entered a " + n + "-digit integer");
  Console.ReadKey();
}
```

4. Solution

```
static void Main() {
  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 / 0.72;
    Console.WriteLine("$" + usd + " = " + eur + " EUR");
  }
  else if (ch == 2) {
    gbp = usd / 0.60;
    Console.WriteLine("$" + usd + " = " + gbp + " GBP");
  }
  else if (ch == 3) {
    jpy = usd / 102.15;
    Console.WriteLine("$" + usd + " = " + jpy + " JPY");
  }
  else {
    cad = usd / 1.10;
    Console.WriteLine("$" + usd + " = " + cad + " CAD");
  }
  Console.ReadKey();
}
```

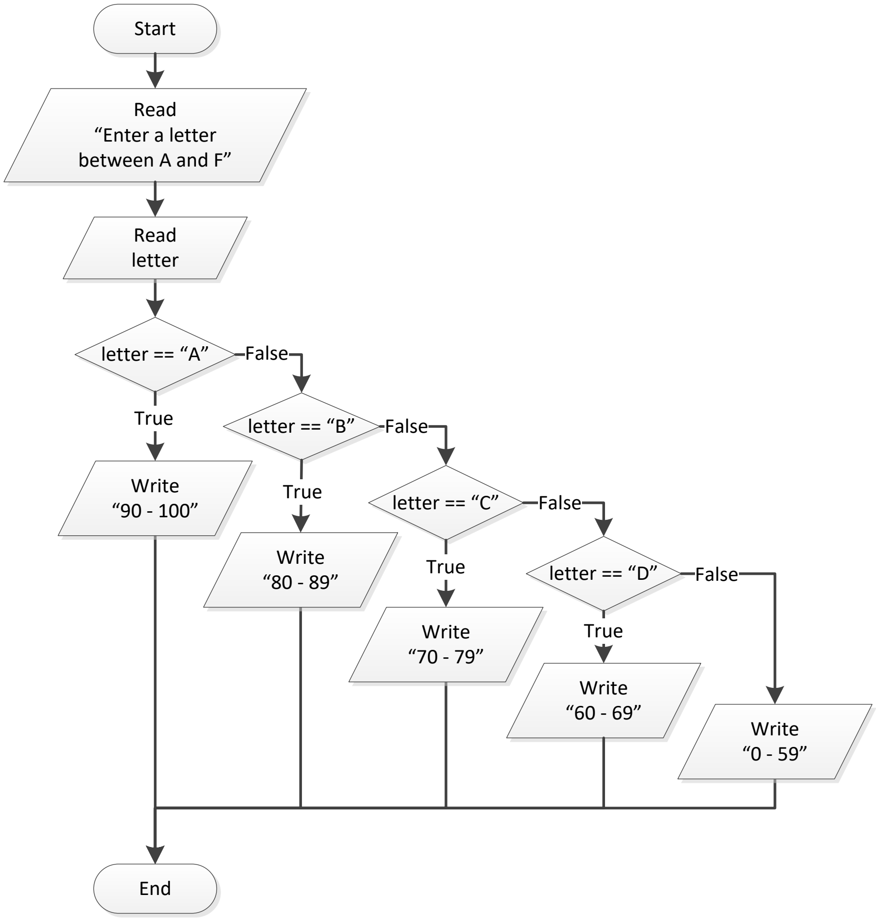
5. Solution

```
static void Main() {
  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)");
  }
  Console.ReadKey();
}
```

6. Solution

```
static void Main() {
  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");
  }
  Console.ReadKey();
}
```

7. Solution



```
static void Main() {
  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");
  }
  Console.ReadKey();
}
```

Chapter 19

19.2 Answers of Review Questions: True/False

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

19.3 Answers of 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.Write(x + ", " + y)`	5, 5 is displayed		

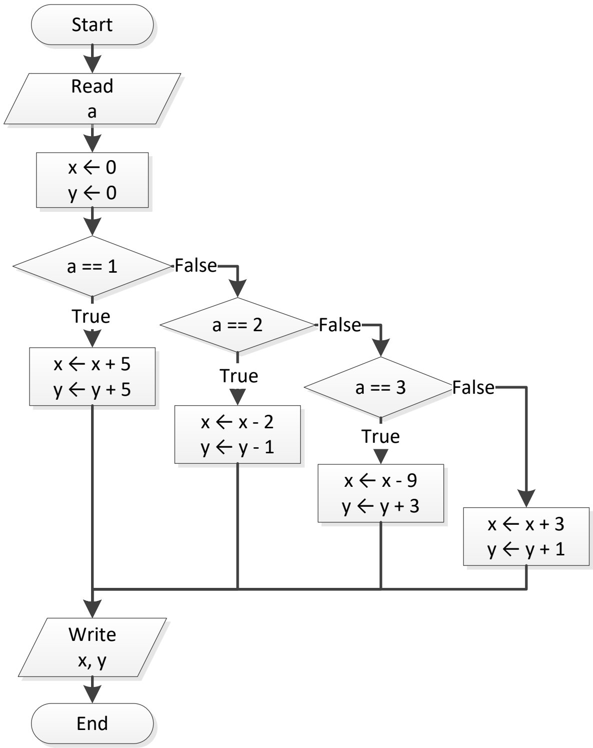
For input value of 3

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.Write(x + ", " + y)`	-9, 3 is displayed		

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	`x = x + 3`	250	3	0
8	`y++`	250	3	1
9	`Console.Write(x + ", " + y)`	3, 1 is displayed		



2. *Solution*

For input values of 10, 2, 5

Step	Statement	a	x	y
1	`a = Int32.Parse(Console.ReadLine())`	**10**	?	?
2	`x = Int32.Parse(Console.ReadLine())`	10	**2**	?
3	`y = Double.Parse(Console.ReadLine())`	10	2	**5**
4	`case a == 10`	True		
5	`x = x % 2`	10	**0**	5
6	`y = Math.Pow(y, 2)`	10	0	**25**
7	`Console.Write(x + ", " + y)`	0, 25 is displayed		

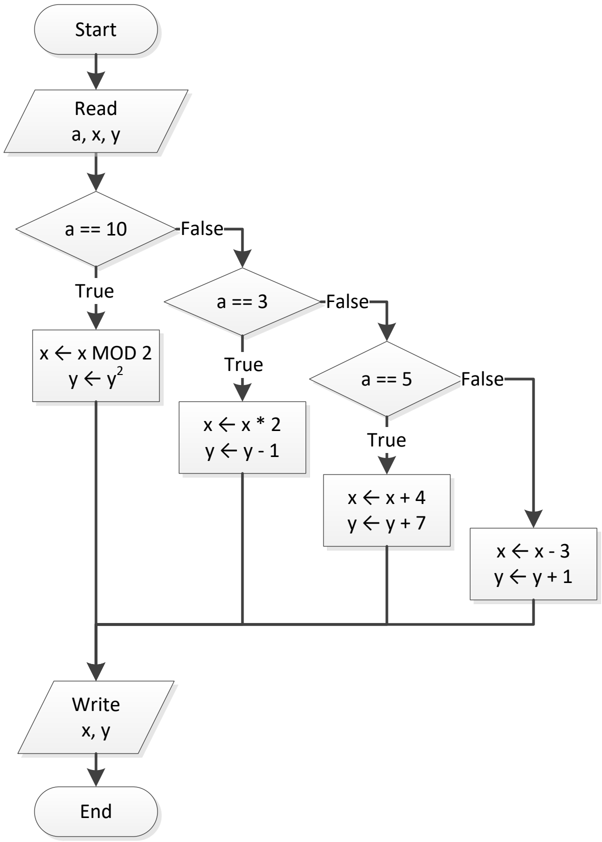
For input values of 5, 2, 3

Step	Statement	a	x	y
1	`a = Int32.Parse(Console.ReadLine())`	**5**	?	?
2	`x = Int32.Parse(Console.ReadLine())`	5	**2**	?

3	`y = Double.Parse(Console.ReadLine())`	5	2	**3**
4	`case a == 10`	False		
5	`case a == 3`	False		
6	`case a == 5`	True		
7	`x = x + 4`	5	**6**	3
8	`y += 7`	5	6	**10**
9	`Console.Write(x + ", " + y)`	6, 10 is displayed		

For input values of 4, 6, 2

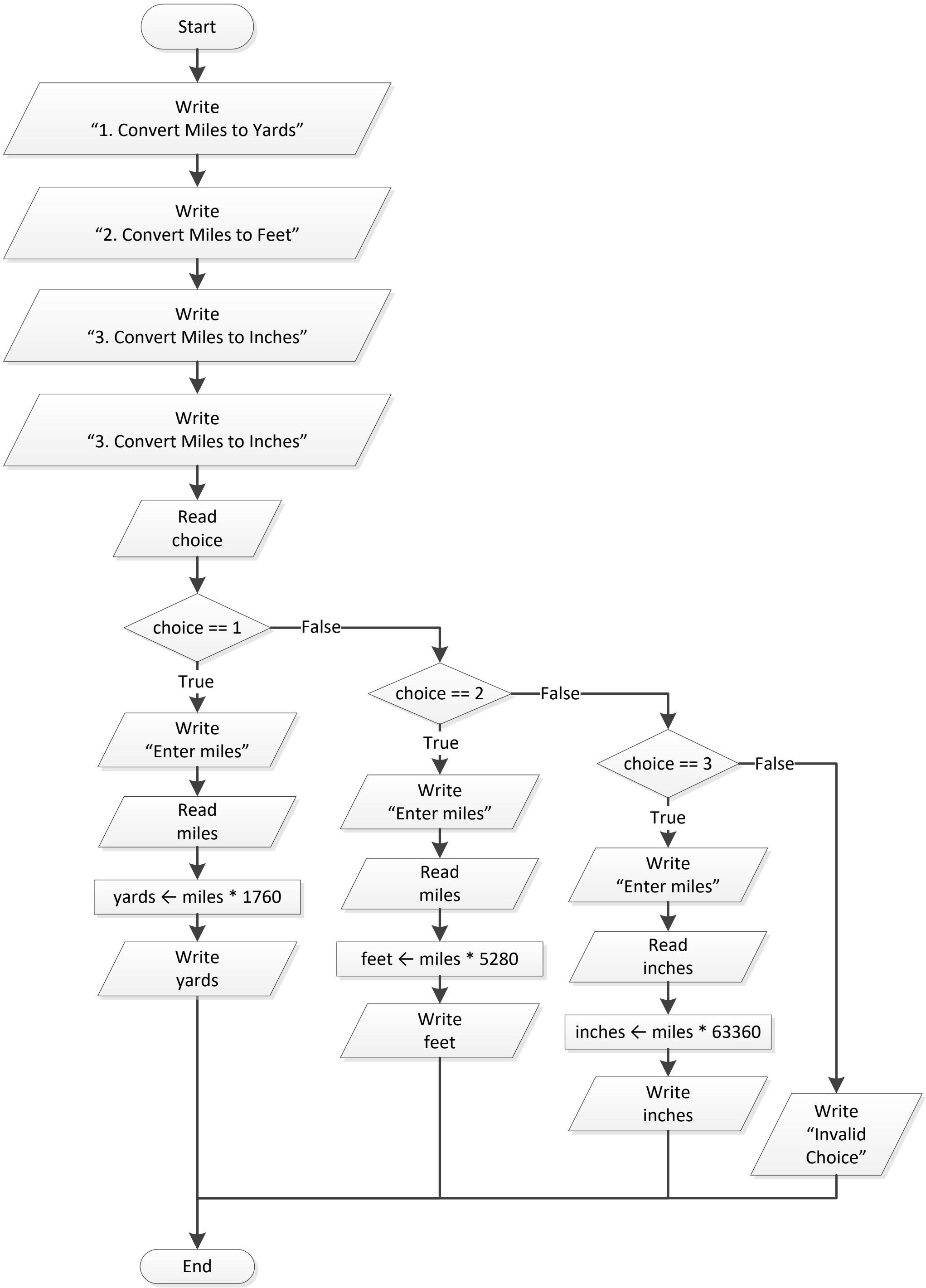
Step	Statement	a	x	y
1	`a = Int32.Parse(Console.ReadLine())`	**4**	?	?
2	`x = Int32.Parse(Console.ReadLine())`	4	**6**	?
3	`y = Double.Parse(Console.ReadLine())`	4	6	**2**
4	`case a == 10`	False		
5	`case a == 3`	False		
6	`case a == 5`	False		
7	`x -= 3`	4	**3**	2
8	`y++`	4	3	**3**
9	`Console.Write(x + ", " + y)`	3, 3 is displayed		



3. Solution

```
static void Main() {
  int number;

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

  switch (number) {
    case 1:
      Console.WriteLine("JANUARY");
      break;
    case 2:
      Console.WriteLine("FEBRUARY");
      break;
    case 3:
      Console.WriteLine("MARCH");
      break;
    case 4:
      Console.WriteLine("APRIL");
      break;
    case 5:
      Console.WriteLine("MAY");
      break;
    case 6:
      Console.WriteLine("JUNE");
      break;
    case 7:
      Console.WriteLine("JULY");
      break;
    case 8:
      Console.WriteLine("AUGUST");
      break;
    case 9:
      Console.WriteLine("SEPTEMBER");
      break;
    case 10:
      Console.WriteLine("OCTOBER");
      break;
    case 11:
      Console.WriteLine("NOVEMBER");
      break;
    case 12:
      Console.WriteLine("DECEMBER");
      break;
    default:
      Console.WriteLine("Error");
      break;
  }
  Console.ReadKey();
}
```

4. Solution



```
static void Main() {
  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;
  }
  Console.ReadKey();
}
```

5. Solution

```
static void Main() {
  string roman;

  Console.Write("Enter a roman number 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;
  }
  Console.ReadKey();
}
```

6. Solution

```
static void Main() {
  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;
  }
  Console.ReadKey();
}
```

7. Solution

```
static void Main() {
  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;
  }
  Console.ReadKey();
}
```

8. Solution

```
static void Main() {
  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;
  }
  Console.ReadKey();
}
```

9. Solution

```
static void Main() {
  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;
  }
  Console.ReadKey();
}
```

Chapter 20

20.3 Answers of Review Questions: True/False

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

20.4 Answers of Review Exercises

1. Solution

For input values of 20, 1

Step	Statement	x	y
1	`x = Int32.Parse(Console.ReadLine())`	**20**	?
2	`y = Int32.Parse(Console.ReadLine())`	20	**1**
3	`if (x < 30)`	True	
4	`case y == 1`	True	
5	`x = x % 3`	**2**	1
6	`y = 5`	2	**5**
7	`Console.Write(x + ", " + y)`	2, 5 is displayed	

For input values of 20, 3

Step	Statement	x	y
1	`x = Int32.Parse(Console.ReadLine())`	**20**	?
2	`y = Int32.Parse(Console.ReadLine())`	20	**3**
3	`if (x < 30)`	True	
4	`case y == 1`	False	
5	`case y == 2`	False	
6	`case y == 3`	True	
7	`x = x + 5`	**25**	3
8	`y += 3`	25	**6**
9	`Console.Write(x + ", " + y)`	25, 6 is displayed	

For input values of 12, 8

Step	Statement	x	y
1	`x = Int32.Parse(Console.ReadLine())`	**12**	?
2	`y = Int32.Parse(Console.ReadLine())`	12	**8**
3	`if (x < 30)`	True	
4	`case y == 1`	False	
5	`case y == 2`	False	

6	`case y == 3`	False	
7	`x -= 2`	**10**	8
8	`y++`	10	**9**
9	`Console.Write(x + ", " + y)`	10, 9 is displayed	

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	`y++`	50	**1**
4	`Console.Write(x + ", " + y)`	50, 1 is displayed	

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.Write(x + ", " + y)`	125, 26 is displayed	

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.Write(x + ", " + y)`	2, 2 is displayed	

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	`x = x * 5`	**100**	15
7	`y += 2`	100	**17**
8	`Console.Write(x + ", " + y)`	100, 17 is displayed	

For input values of 10, 30

Step	Statement	x	y
1	`x = Int32.Parse(Console.ReadLine())`	**10**	?
2	`y = Int32.Parse(Console.ReadLine())`	10	30
3	`if ((x + y) / 2 <= 20)`	True	
4	`if (y < 10)`	False	
5	`else if (y < 20)`	False	
6	`x = x - 2`	**8**	30
7	`y += 3`	8	**33**
8	`Console.Write(x + ", " + y)`	8, 33 is displayed	

3. Solution

```
static void Main() {
  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");
    }
  }
  Console.ReadKey();
}
```

4. Solution

```
static void Main() {
  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 + " notes of $10 " + usd5 + " notes of $5 ");
    Console.WriteLine("and " + usd1 + " notes of $1");
  }
  Console.ReadKey();
}
```

5. Solution

First Approach

```
static void Main() {
  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");
    }
  }
  Console.ReadKey();
}
```

Second Approach

```
static void Main() {
  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.Write("The day is " + message1 + " and " + message2);
  Console.ReadKey();
}
```

Chapter 21

21.8 Answers of Review Questions: True/False

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

21.9 Answers of Review Questions: Multiple Choice

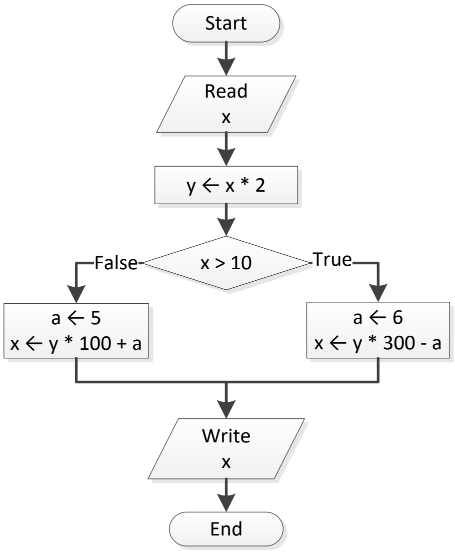
1. a
2. b
3. a
4. d

21.10 Answers of Review Exercises

1. Solution

```
static void Main() {
  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.Write(a);
  Console.ReadKey();
}
```

2. Solution



3. Solution

```
static void Main() {
  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);
  }
  Console.ReadKey();
}
```

4. Solution

```
static void Main() {
  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.Write(y + ", " + b);
  Console.ReadKey();
}
```

5. Solution

```
static void Main() {
  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.Write(y);
  Console.ReadKey();
}
```

6. Solution

```
static void Main() {
  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");
      }
    }
  }
  Console.ReadKey();
}
```

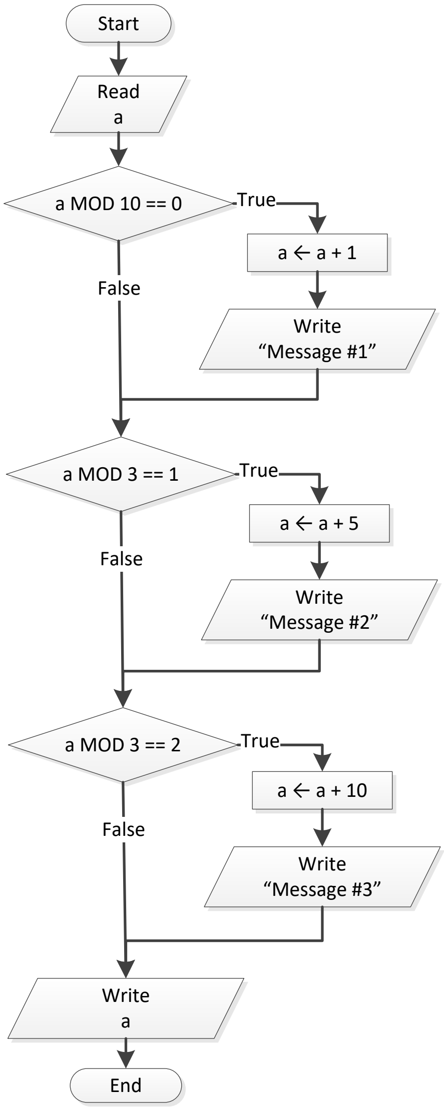
7. Solution

```
static void Main() {
  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!");
  }
  Console.ReadKey();
}
```

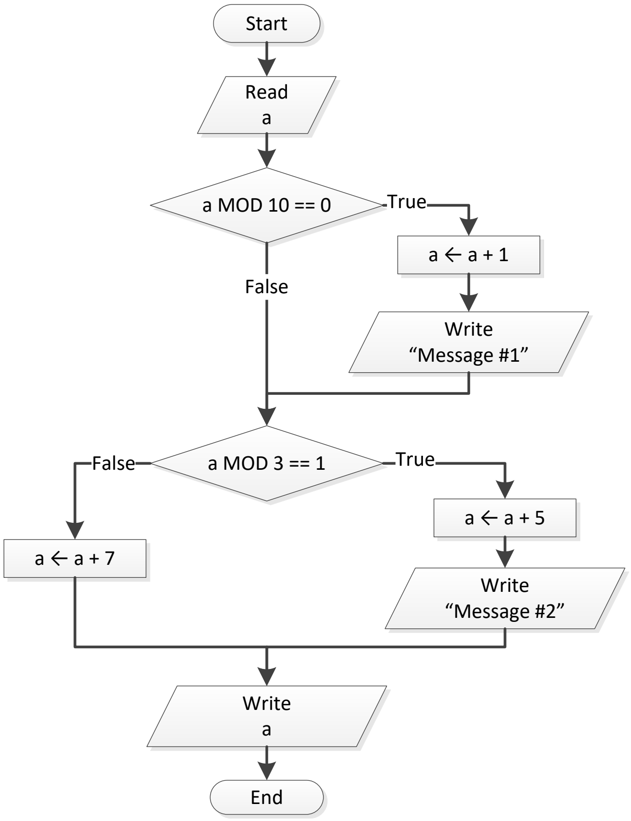
Chapter 22

22.4 Answers of Review Exercises

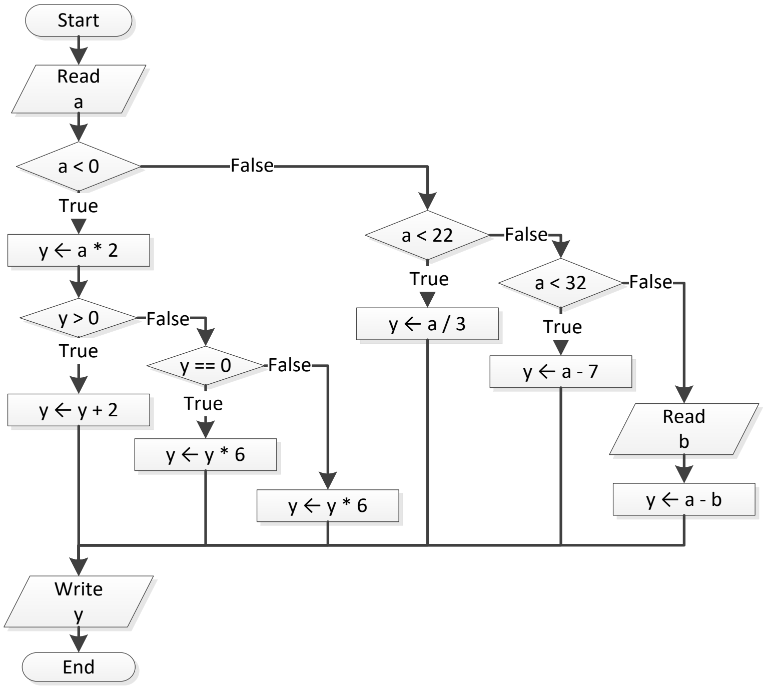
1. *Solution*



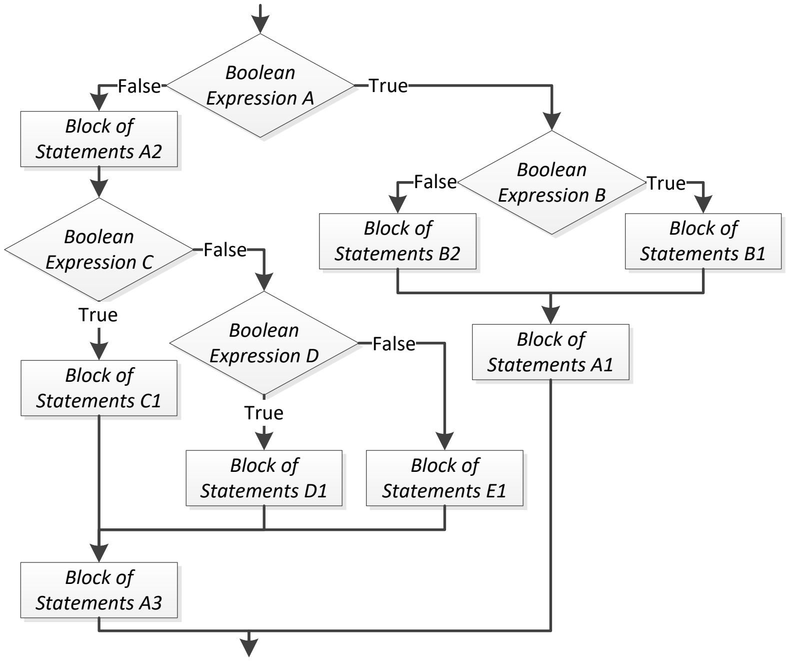
2. Solution



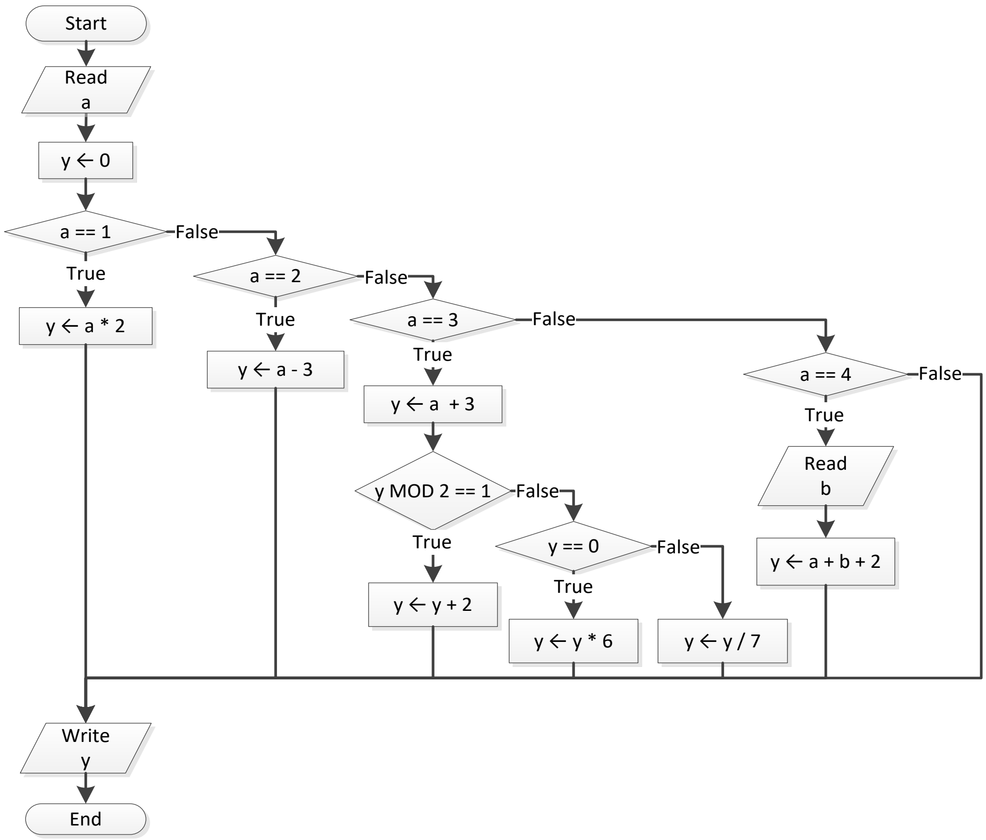
3. Solution



4. Solution



5. Solution



6. Solution

```
static void Main() {
  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.Write(x + ", " + y);
  Console.ReadKey();
}
```

7. Solution

```
static void Main() {
  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");
  }
  Console.ReadKey();
}
```

8. Solution

```
static void Main() {
  int x;
  string x_str;

  x_str = Console.ReadLine();
  if (Int32.TryParse(x_str, out x) == true) {
    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_str == "Exit") {
      Console.WriteLine("Bye");
    }
    else {
      Console.WriteLine("Invalid Number");
    }
  }
  Console.ReadKey();
}
```

9. Solution

```
static void Main() {
  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;
    }
  }
  Console.ReadKey();
}
```

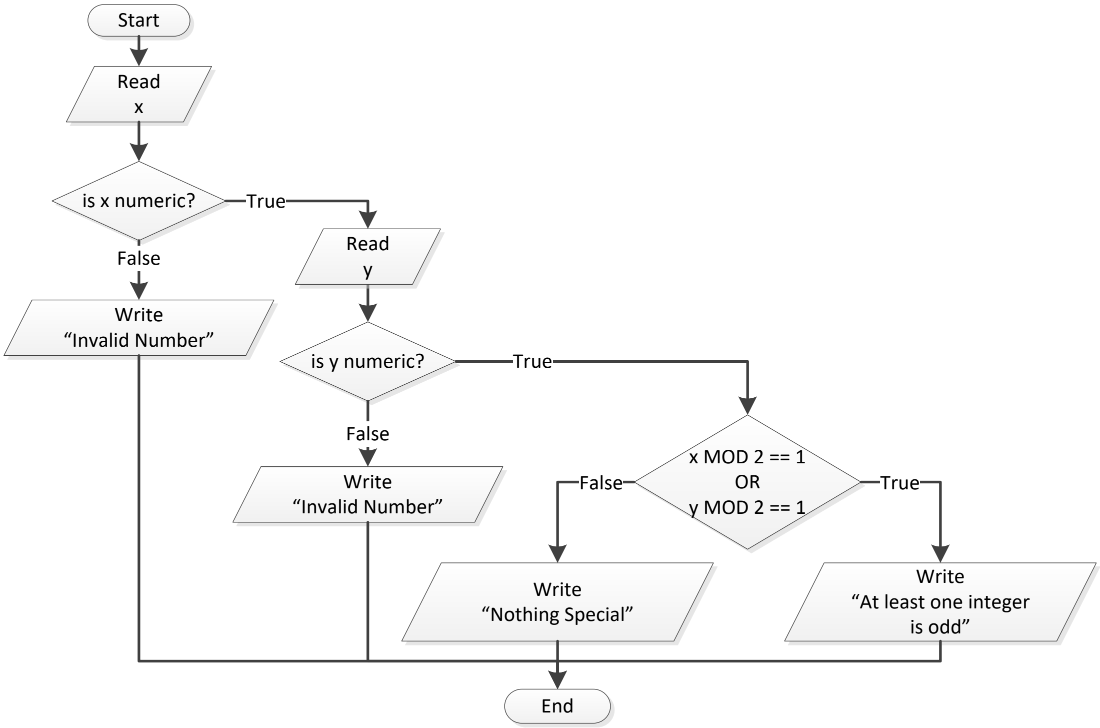
10. Solution

```
static void Main() {
  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.Write(a + ", " + b  + ", " + c);
  Console.ReadKey();
}
```

Chapter 23

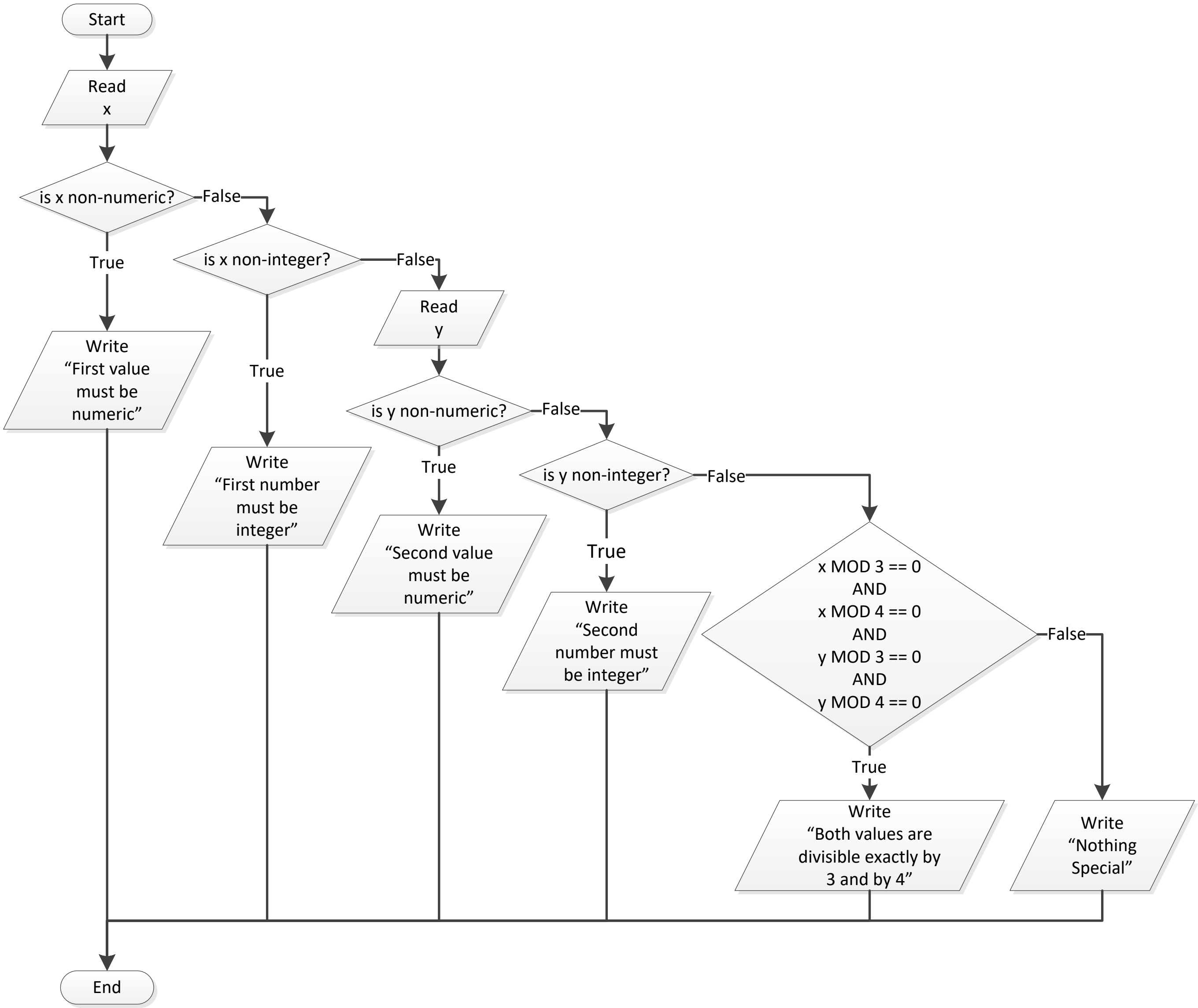
23.5 Answers of Review Exercises

1. Solution



```
static void Main() {
  int x, y;
  string input;

  input = Console.ReadLine();

  if (Int32.TryParse(input, out x) == true) {
    input = Console.ReadLine();
    if (Int32.TryParse(input, out y) == true) {
      if (x % 2 == 1 || y % 2 == 1) {
        Console.WriteLine("At least one integer is odd");
      }
      else {
        Console.WriteLine("Nothing Special");
      }
    }
    else {
      Console.WriteLine("Invalid Number");
    }
  }
  else {
    Console.WriteLine("Invalid Number");
  }
  Console.ReadKey();
}
```

2. Solution



```
static void Main() {
  double x, y;
  string input;

  input = Console.ReadLine();

  if (Double.TryParse(input, out x) == false) {
    Console.WriteLine("First value must be numeric");
  }
  else if (x != (int)x) {
    Console.WriteLine("First number must be integer");
  }
  else {
    input = Console.ReadLine();
    if (Double.TryParse(input, out y) == false) {
      Console.WriteLine("Second value must be numeric");
    }
    else if (y != (int)y) {
      Console.WriteLine("Second number must be integer");
    }
    else {
      if ((int)x % 3 == 0 && (int)x % 4 == 0 &&
          (int)y % 3 == 0 && (int)y % 4 == 0) {

        Console.WriteLine("Both values are divisible exactly by 3 and by 4");
      }
      else {
        Console.WriteLine("Nothing Special");
      }
    }
  }
  Console.ReadKey();
}
```

3. Solution

```
static void Main() {
  int choice;
  double t;
  string input;

  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: ");
  input = Console.ReadLine();

  if (choice < 1 || choice > 4) {
    Console.WriteLine("Wrong choice");
  }
  else if (Double.TryParse(input, out t) == false) {
    Console.WriteLine("Wrong temperature");
  }
  else {
    switch (choice) {
      case 1:
        Console.WriteLine(1.8 * t - 459.67);
        break;
      case 2:
        Console.WriteLine((t + 459.57) / 1.8);
        break;
      case 3:
        Console.WriteLine(5 / 9 * (t - 32));
        break;
      case 4:
        Console.WriteLine(9 / 5 * t + 32);
        break;
    }
  }
  Console.ReadKey();
}
```

4. Solution

```
static void Main() {
  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());

  if (op == "+")
    Console.WriteLine(a + b);
  else if (op == "-")
    Console.WriteLine(a - b);
  else if (op == "*")
    Console.WriteLine(a * b);
  else if (op == "/") {
    if (b == 0) {
      Console.WriteLine("Error: Division by zero");
    }
    else {
      Console.WriteLine(a / (double)b);
    }
  }
  else if (op == "DIV") {
    if (b == 0) {
      Console.WriteLine("Error: Division by zero");
    }
    else {
      Console.WriteLine((int)(a / b));
    }
  }
  else if (op == "MOD") {
    if (b == 0) {
      Console.WriteLine("Error: Division by zero");
    }
    else {
      Console.WriteLine(a % b);
    }
  }
  else if (op == "POWER")
    Console.WriteLine(Math.Pow(a, b));

  Console.ReadKey();
}
```

5. Solution

```
static void Main() {
  int age1, age2, age3, max, middle, min;

  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());

  min = age1;
  if (age2 < min) {
    min = age2;
  }
  if (age3 < min) {
    min = age3;
  }
  max = age1;
  if (age2 > max) {
    max = age2;
  }
  if (age3 > max) {
    max = age3;
  }

  middle = age1 + age2 + age3 - min - max;
  Console.Write(middle);
  Console.ReadKey();
}
```

6. Solution

```
static void Main() {
  int a1, a2, a3, max, middle, min;
  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();

  min = a1;
  min_name = n1;
  if (a2 > min) {
    min = a2;
    min_name = n2;
  }
  if (a3 > min) {
    min = a3;
    min_name = n3;
  }

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

  middle = a1 + a2 + a3 - min - max;

  if (Math.Abs(max - middle) < Math.Abs(min - middle)) {
    Console.WriteLine(max_name);
  }
  else {
    Console.WriteLine(min_name);
  }
  Console.ReadKey();
}
```

7. Solution

```
static void Main() {
  int digit1, digit2, digit3, r, x, sum;
  string input;

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

  if (Int32.TryParse(input, out x) == false) {
    Console.WriteLine("Entered value contains non-numeric characters");
  }
  else if (x < 100 || x > 999) {
    Console.WriteLine("Entered integer is not a three-digit integer");
  }
  else {
    digit1 = (int)(x / 100);
    r = x % 100;

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

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

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

8. Solution

```
static void Main() {
  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);
  }
  Console.ReadKey();
}
```

9. Solution

First Approach

```
static void Main() {
  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).ToLower() +
          word.Substring(4, 1).ToUpper() +
          word.Substring(5, 1).ToLower();

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

Second Approach

```
static void Main() {
  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().ToLower() +
          word[4].ToString().ToUpper() +
          word[5].ToString().ToLower();

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

10. Solution

```
static void Main() {
  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.Write("You must pay $" + payment);
  Console.ReadKey();
}
```

11. Solution

```
const double VAT = 0.19;

static void Main() {
  double amount, discount, payment;
  string input;

  Console.Write("Enter total amount: ");
  input = Console.ReadLine();

  if (Double.TryParse(input, out amount) == false) {
    Console.WriteLine("Entered value contains non-numeric characters");
  }
  else if (amount < 0) {
    Console.WriteLine("Entered non-negative value");
  }
  else {
    if (amount < 50) {
      discount = 0;
    }
    else if (amount < 100) {
      discount = 1;
    }
    else if (amount < 250) {
      discount = 2;
    }
    else {
      discount = 3;
    }

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

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

12. Solution

```
static void Main() {
  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");
    }
  }
  Console.ReadKey();
}
```

13. Solution

```
const double TAX_RATE = 0.10;

static void Main() {
  int water;
  double total;
  string input;

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

  if (Int32.TryParse(input, out water) == false) {
    Console.WriteLine("Entered value contains non-numeric characters");
  }
  else if (water < 0) {
    Console.WriteLine("Entered value is negative");
  }
  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);
  }
  Console.ReadKey();
}
```

14. Solution

```
static void Main() {
  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.Write("Tax: " + tax);
  Console.ReadKey();
}
```

15. Solution

```
static void Main() {
  double wind;
  string input;

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

  if (Double.TryParse(input, out wind) == false) {
    Console.WriteLine("Entered value contains non-numeric characters");
  }
  else if (wind < 0) {
    Console.WriteLine("Entered value is negative");
  }
  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!!!");
    }
  }
  Console.ReadKey();
}
```

Chapter 24

24.3 Answers of Review Questions: True/False

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

Chapter 25

25.2 Answers of Review Questions: True/False

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

25.3 Answers of Review Questions: Multiple Choice

1. c
2. b
3. d
4. b
5. c
6. b

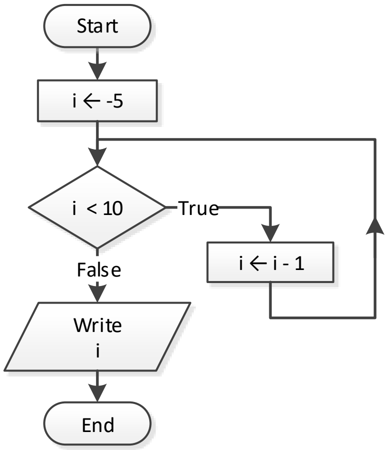
25.4 Answers of Review Exercises

1. 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.Write(x)`	2 is displayed	

It performs 3 iterations

2. 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

3. 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)	3, 6 is displayed			
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)	6, 7 is displayed			
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)	10, 11, 12 is displayed			
26	a += 4	**14**	11	22	12
27	while (a <= 10)	False			

4. *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				

5. Solution

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

6. Solution

```
static void Main() {
  double a, sum;
  int i, n;

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

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

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

7. Solution

```
static void Main() {
  int a, i, n, p;

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

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

8. Solution

```
static void Main() {
  int a, i, sum;

  sum = 0;

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

9. Solution

```
static void Main() {
  int a, i, sum;

  sum = 0;

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

10. Solution

```
static void Main() {
  double a, p;

  p = 1;

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

Step	Statement	a	p
1	`p = 1`	?	**1**
2	`a = Double.Parse(Console.ReadLine())`	**3**	1
3	`while (a != 0)`	True	
4	`p = p * a`	3	**3**
5	`a = Double.Parse(Console.ReadLine())`	**2**	3
6	`while (a != 0)`	True	
7	`p = p * a`	2	**6**
8	`a = Double.Parse(Console.ReadLine())`	**9**	6
9	`while (a != 0)`	True	
10	`p = p * a`	9	**54**
11	`a = Double.Parse(Console.ReadLine())`	**0**	54
12	`while (a != 0)`	False	
13	`Console.Write(p)`	54 is displayed	

11. Solution

```
static void Main() {
  int years;
  double population;

  population = 30000;

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

Chapter 26

26.2 Answers of Review Questions: True/False

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

26.3 Answers of Review Questions: Multiple Choice

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

26.4 Answers of Review Exercises

1. Solution

```
static void Main() {
  int i;

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

2. 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.Write(y)`	10 is displayed	
6	`while (y < x)`	True	
7	`y *= 2`	39	**20**
8	`x++`	**40**	20
9	`Console.Write(y)`	20 is displayed	
10	`while (y < x)`	True	
11	`y *= 2`	40	**40**
12	`x++`	**41**	40
13	`Console.Write(y)`	40 is displayed	
14	`while (y < x)`	True	

15	y *= 2	41	**80**
16	x++	**42**	80
17	Console.Write(y)	80 is displayed	
18	while (y < x)	False	

3. Solution

Step	Statement	Notes	x
1	x = 1		**1**
2	if (x % 2 == 0)	False	
3	x += 3		**4**
4	Console.Write(x)	4 is displayed	
5	while (x < 12)	True	
6	if (x % 2 == 0)	True	
7	x++		**5**
8	Console.Write(x)	5 is displayed	
9	while (x < 12)	True	
10	if (x % 2 == 0)	False	
11	x += 3		**8**
12	Console.Write(x)	8 is displayed	
13	while (x < 12)	True	
14	if (x % 2 == 0)	True	
15	x++		**9**
16	Console.Write(x)	9 is displayed	
17	while (x < 12)	True	
18	if (x % 2 == 0)	False	
19	x += 3		**12**
20	Console.Write(x)	12 is displayed	
21	while (x < 12)	False	

4. Solution

Step	Statement	x	y
1	y = 2	?	**2**
2	x = 0	**0**	2
3	y = Math.Pow(y, 2)	0	**4**
4	if (x < 256)	True	
5	x = x + y	**4**	

6	Console.Write(x + ", " + y)	4, 4 is displayed	
7	while (y < 65535)	True	
8	y = Math.Pow(y, 2)	4	**16**
9	if (x < 256)	True	
10	x = x + y	**20**	16
11	Console.Write(x + ", " + y)	20, 16 is displayed	
12	while (y < 65535)	True	
13	y = Math.Pow(y, 2)	20	**256**
14	if (x < 256)	True	
15	x = x + y	**276**	256
16	Console.Write(x + ", " + y)	276, 256 is displayed	
17	while (y < 65535)	True	
18	y = Math.Pow(y, 2)	276	**65536**
19	if (x < 256)	False	
20	Console.Write(x + ", " + y)	276, 65536 is displayed	
21	while (y < 65535)	False	

5. Solution

Step	Statement	a	b	c	d	x
1	a = 2	**2**	?	?	?	?
2	b = 4	2	**4**	?	?	?
3	c = 0	2	4	**0**	?	?
4	d = 0	2	4	0	**0**	?
5	x = a + b	2	4	0	0	**6**
6	if (x % 2 != 0)	False				
7	else if (d % 2 == 0)	True				
8	d = d + 5	2	4	0	**5**	6
9	a = b	**4**	4	0	5	6
10	b = d	4	**5**	0	5	6
11	while (c < 11)	True				
12	x = a + b	4	5	0	5	**9**
13	if (x % 2 != 0)	True				
14	c = c + 5	4	5	**5**	5	9

15	`a = b`	**b**	5	5	5	9
16	`b = d`	5	**5**	5	5	9
17	`while (c < 11)`	True				
18	`x = a + b`	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	`x = a + b`	5	5	8	5	**10**
25	`c = c + 3`	5	5	**11**	5	10
26	`a = b`	**5**	5	11	5	10
27	`b = d`	5	**5**	11	5	10
28	`while (c < 11)`	False				

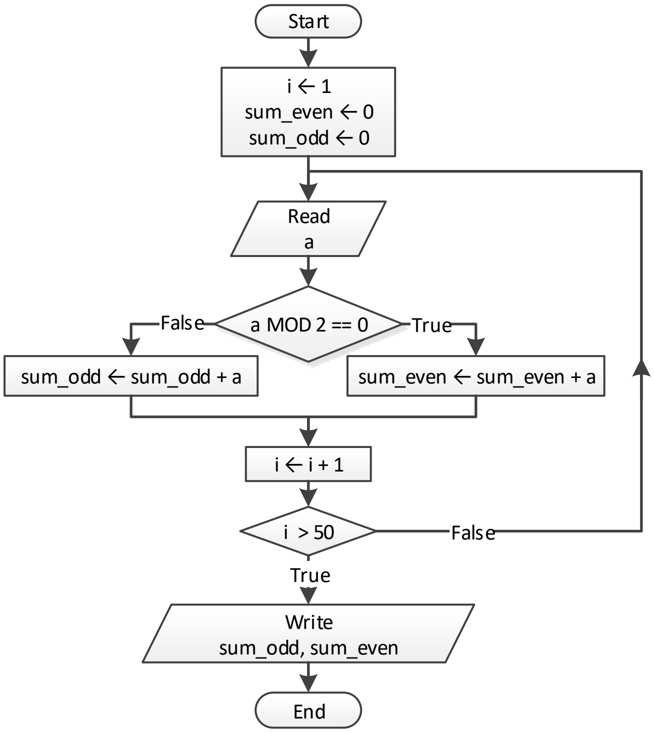
6. Solution

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

7. Solution

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

8. Solution



```
static void Main() {
  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.Write(sum_even + ", " + sum_odd);
  Console.ReadKey();
}
```

9. Solution

```
static void Main() {
  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.Write(Math.Abs(p));
  Console.ReadKey();
}
```

10. Solution

```
static void Main() {
  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.Write(p);
  Console.ReadKey();
}
```

11. Solution

```
static void Main() {
  int a, sum;

  sum = 0;

  a = Int32.Parse(Console.ReadLine());
  if (a > 0) {
    do {
      sum = sum + a;
      a = Int32.Parse(Console.ReadLine());
    } while (a > 0);
  }
  Console.Write(sum);
  Console.ReadKey();
}
```

Step	Statement	a	sum
1	`sum = 0`	?	**0**
2	`a = Int32.Parse(Console.ReadLine())`	**5**	0
3	`if (a > 0)`	True	

4	`sum = sum + a`	5	**5**
5	`a = Int32.Parse(Console.ReadLine())`	**2**	5
6	`while (a > 0)`	True	
7	`sum = sum + a`	2	**7**
8	`a = Int32.Parse(Console.ReadLine())`	**3**	7
9	`while (a > 0)`	True	
10	`sum = sum + a`	3	**10**
11	`a = Int32.Parse(Console.ReadLine())`	**0**	10
12	`while (a > 0)`	False	

12. Solution

```
static void Main() {
  double population;
  int years;

  population = 50000;

  years = 0;
  while (population >= 20000) {
    population -= population * 0.10;
    years++;
  }
  Console.Write(years);
  Console.ReadKey();
}
```

Chapter 27

27.3 Answers of Review Questions: True/False

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

27.4 Answers of Review Questions: Multiple Choice

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

27.5 Answers of 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.Write(a + ", " + b)`	12, 3 is displayed		

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.Write(b)`	24 is displayed		

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.Write(b)`	1 is displayed		

3. *Solution*

For input value of 12

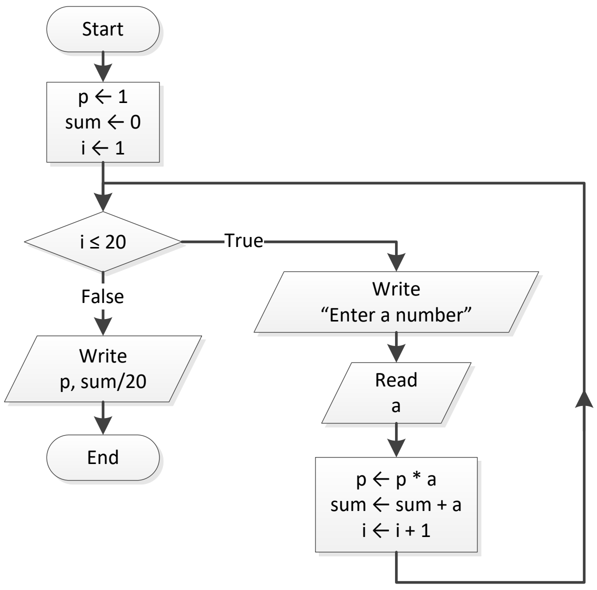
Step	Statement	a	x	y	j
1	`a = Int32.Parse(Console.ReadLine())`	**12**	?	?	?
2	`j = 2`	12	?	?	**2**
3	`j <= a - 1`	True			
4	`x = j * 3 + 3`	12	**9**	?	2
5	`y = j * 2 + 10`	12	9	**14**	2
6	`if (y - x > 0 || x > 30)`	True			
7	`y *= 2`	12	9	**28**	2
8	`x += 4`	12	**13**	28	2
9	`Console.Write(x + ", " + y)`	13, 28 is displayed			
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.Write(x + ", " + y)`	22, 40 is displayed			
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.Write(x + ", " + y)`	31, 26 is displayed			
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.Write(x + ", " + y)`	40, 64 is displayed			
33	`j += 3`	12	40	64	**14**
34	`j <= a - 1`	False			

4. *Solution*

i. 9
ii. 2
iii. -7
iv. -1

5. *Solution*

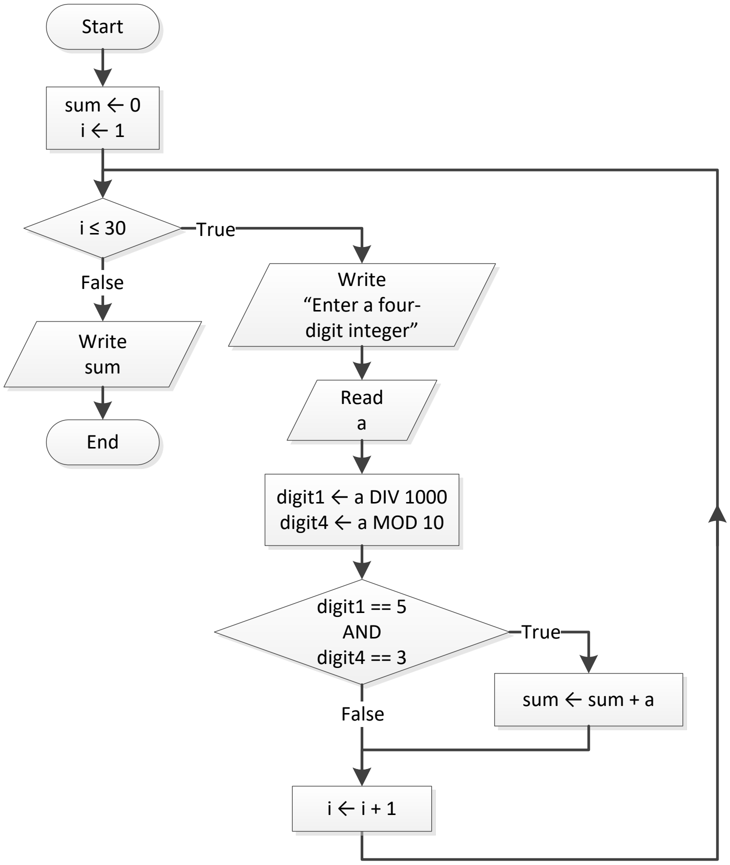


```
static void Main() {
  double a, p, sum;
  int i;
  p = 1;
  sum = 0;
  for (i = 1 ; i <= 20; i++) {
    Console.Write("Enter a number: ");
    a = Double.Parse(Console.ReadLine());
    p = p * a;
    sum = sum + a;
  }
  Console.WriteLine(p);
  Console.Write(sum / 20);
  Console.ReadKey();
}
```

6. Solution

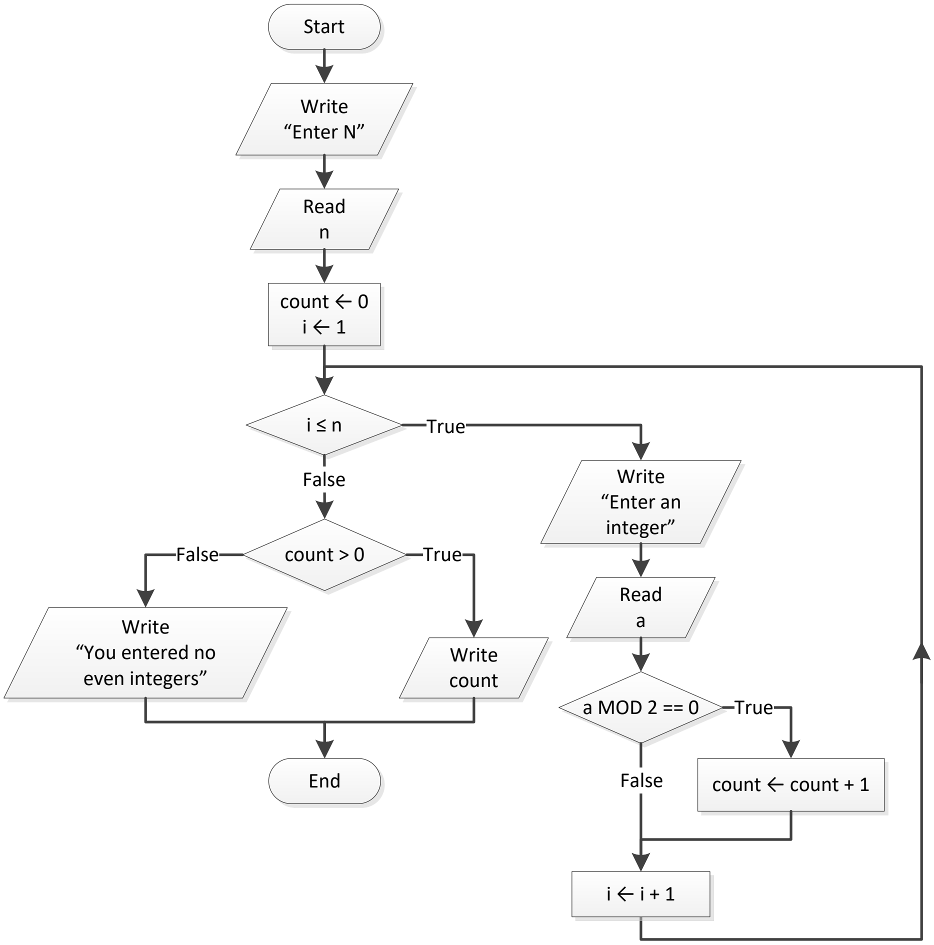
```
static void Main() {
  double deg, i;

  Console.Write("Enter degrees: ");
  deg = Console.ReadLine();
  for (i = 0 ; i <= deg; i += 0.5) {
    Console.WriteLine(Math.Sin(i * Math.PI / 180));
  }
  Console.ReadKey();
}
```

7. Solution



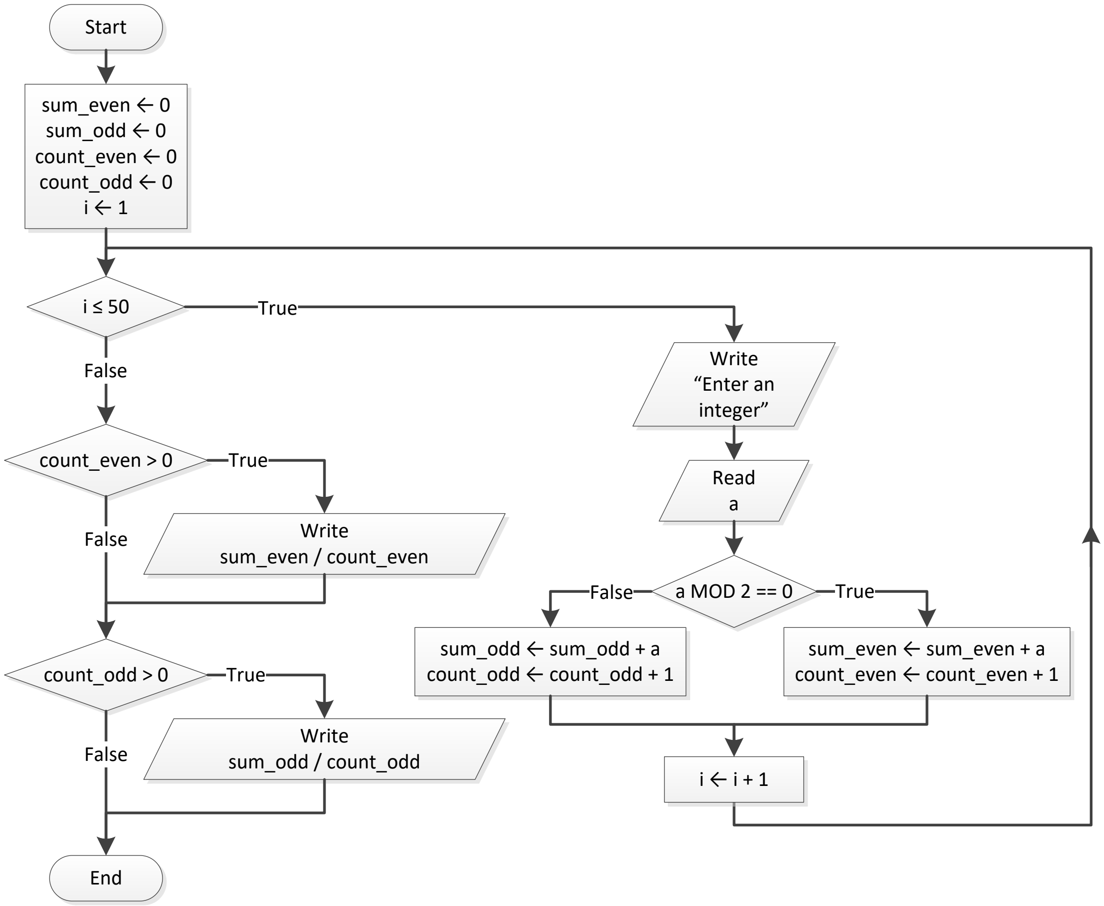
```
static void Main() {
  int a, digit1, digit4, i, sum;

  sum = 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) {
      sum += a;
    }
  }
  Console.Write(sum);
  Console.ReadKey();
}
```

8. Solution



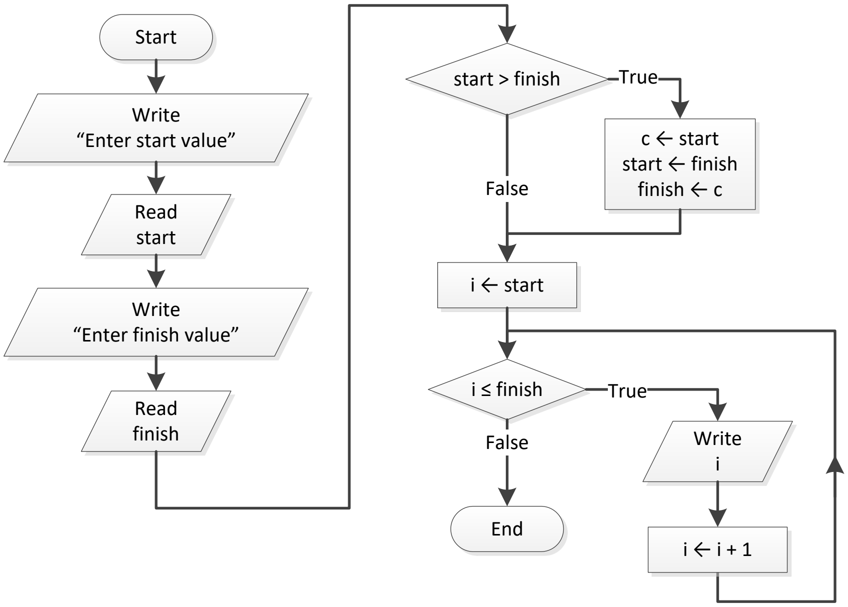
```
static void Main() {
  int a, count, i, n;

  Console.Write("Enter N: ");
  n = Int32.Parse(Console.ReadLine());
  count = 0;
  for (i = 0 ; 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");
  }
  Console.ReadKey();
}
```

9. Solution



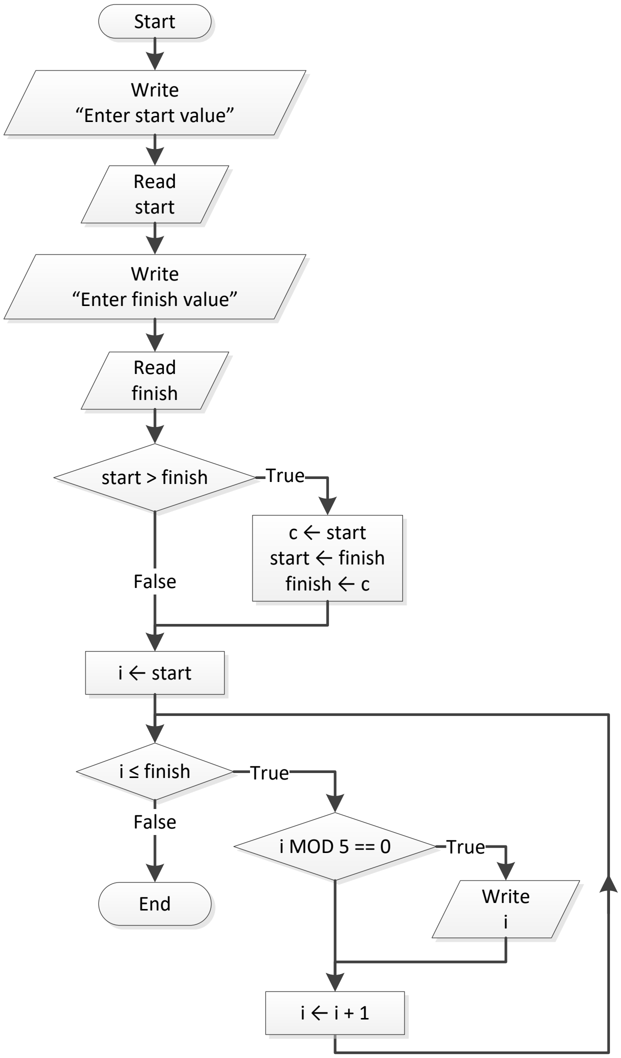
```
static void Main() {
  int a, count_even, count_odd, i, sum_even, sum_odd;

  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);
  }
  Console.ReadKey();
}
```

10. Solution



```
static void Main() {
  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);
  }
  Console.ReadKey();
}
```

11. Solution



```
static void Main() {
  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);
    }
  }
  Console.ReadKey();
}
```

12. Solution

First Approach

```
static void Main() {
  int exp, i;
  double p, base;

  Console.Write("Enter a value for base: ");
  base = 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 *= base;
    }
  }
  else {
    for (i = 1; i <= -exp; i++) {
      p *= 1 / base;
    }
  }
  Console.Write(p);
  Console.ReadKey();
}
```

Second Approach

```
static void Main() {
  int exp, i;
  double p, base;

  Console.Write("Enter a value for base: ");
  base = 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 *= base;
  }
  if (exp < 0) {
    p = 1 / p;
  }
  Console.Write(p);
  Console.ReadKey();
}
```

13. Solution

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

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

  characters = msg.Length;
  count = 0;
  for (i = 0; i <= characters - 1; i++) {
    if (msg.Substring(i, 1) == " ") {
      count++;
    }
  }
  words = count + 1;

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

14. Solution

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

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

  characters = msg.Length;
  count = 0;
  for (i = 0; i <= characters - 1; i++) {
    if (msg.Substring(i, 1) == " ") {
      count++;
    }
  }
  words = count + 1;
  Console.Write("The average number of letters in each word is ");
  Console.Write((characters - count) / (double)words);
  Console.ReadKey();
}
```

Chapter 28

28.3 Answers of Review Questions: True/False

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

28.4 Answers of Review Questions: Multiple Choice

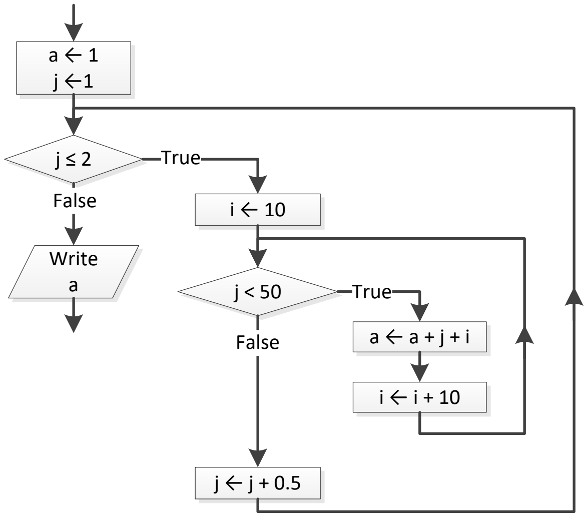
1. b
2. a
3. c
4. a
5. b

28.5 Answers of Review Exercises

1. Solution

i. 10
ii. 4.5
iii. -7
iv. 138

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	`a = a + j + i`	**12**	10	1
7	`i += 10`	12	**20**	1
8	`i < 30`	True		
9	`a = a * j + i`	**33**	20	1
10	`i += 10`	33	**30**	1
11	`i < 30`	False		
12	`j += 0.5`	33	30	**1.5**
13	`j <= 2`	True		
14	`i = 10`	33	**10**	1.5
15	`i < 30`	True		
16	`a = a + j + i`	**44.5**	10	1.5
17	`i += 10`	44.5	**20**	1.5
18	`i < 30`	True		
19	`a = a * j + i`	**66**	20	1.5
20	`i += 10`	66	**30**	1.5
21	`i < 30`	False		
22	`j += 0.5`	66	30	**2**
23	`j <= 2`	True		
24	`i = 10`	66	**10**	2
25	`i < 30`	True		
26	`a = a + j + i`	**78**	10	2
27	`i += 10`	78	**20**	2
28	`i < 30`	True		
29	`a = a * j + i`	**100**	20	2
30	`i += 10`	100	**30**	2
31	`i < 30`	False		
32	`j += 0.5`	100	30	**2.5**
33	`j <= 2`	False		
34	`Console.Write(a)`	100 is displayed		

3. Solution

Step	Statement	s	i	j
1	`s = 0`	**0**	?	?
2	`i = 1`	0	**1**	?
3	`i <= 4`	True		
4	`j = 3`	0	1	**3**

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

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.Write(s)`	61 is displayed			

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	`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	**“NO”**
34	`while (ans == "YES")`	False			
35	`Console.Write(s)`	76 is displayed			

For input values of “YES”, “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	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.Write(s)`	46 is displayed			

5. Solution

```
static void Main() {
  int hour, minutes;

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

6. Solution

```
static void Main() {
  int i, j;

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

7. Solution

```
static void Main() {
  int i, j;

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

8. Solution

```
static void Main() {
  int i, j;
  for (i = 1; i <= 4; i++) {
    for (j = 1; j <= 10; j++) {
      Console.Write("* ");
    }
    Console.WriteLine();
  }
  Console.ReadKey();
}
```

9. Solution

```
static void Main() {
  int i, j, n;

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

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

10. Solution

```
static void Main() {
  int i, j, n;

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

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

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

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

11. Solution

```
static void Main() {
  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();
  }
  Console.ReadKey();
}
```

Chapter 29

29.8 Answers of Review Questions: True/False

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

29.9 Answers of Review Questions: Multiple Choice

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

29.10 Answers of Review Exercises

1. Solution

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

2. Solution

```
static void Main() {
  int denom, i;
  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.Write(s);
  Console.ReadKey();
}
```

3. Solution

```
static void Main() {
  int i, j;

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

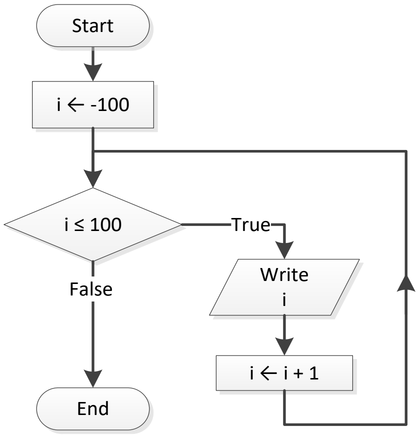
4. Solution

```
static void Main() {
  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();
  }
  Console.ReadKey();
}
```

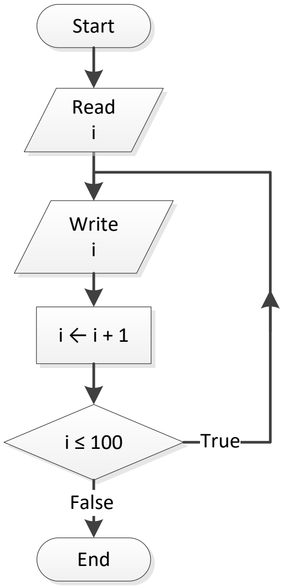
Chapter 30

30.4 Answers of Review Exercises

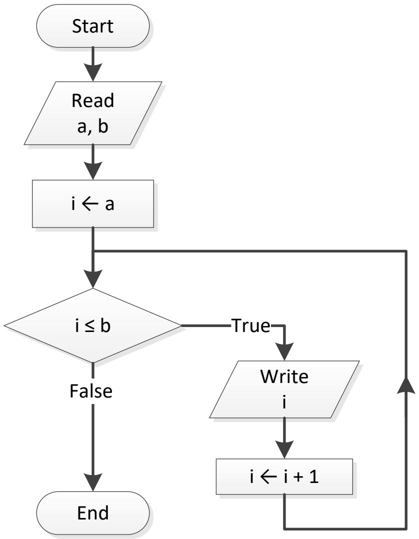
1. Solution



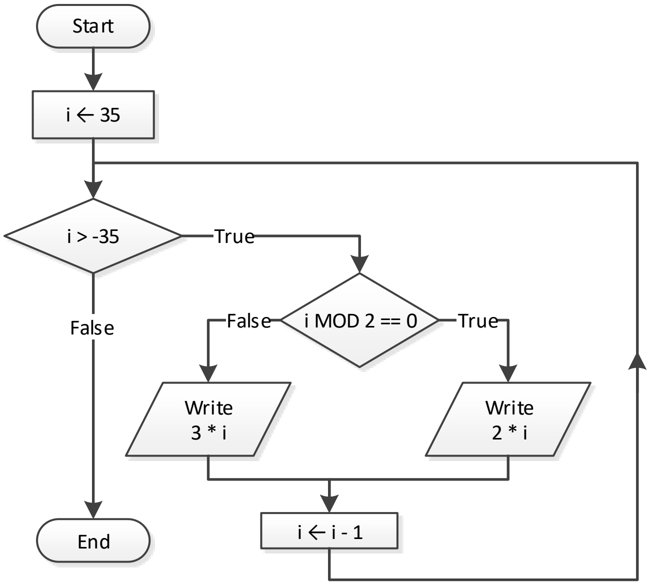
2. Solution



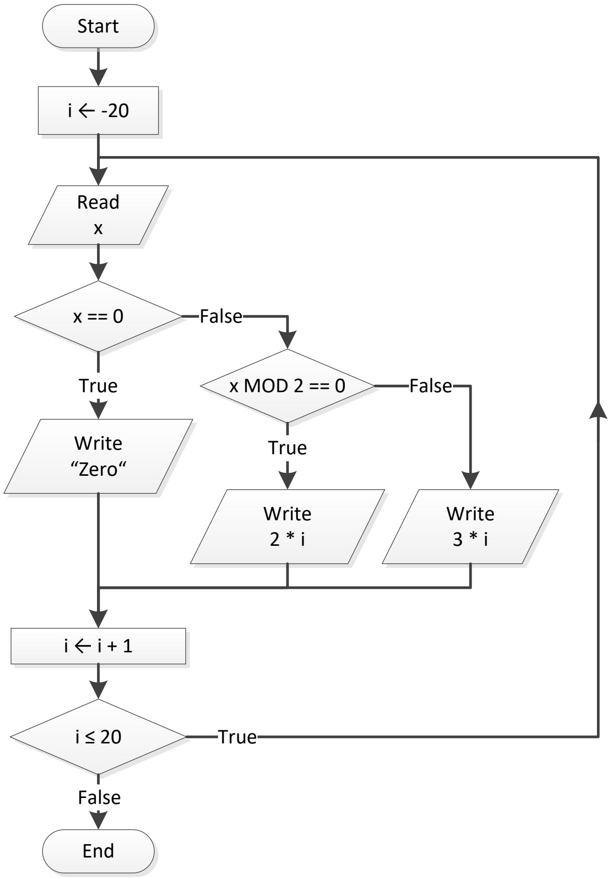
3. Solution



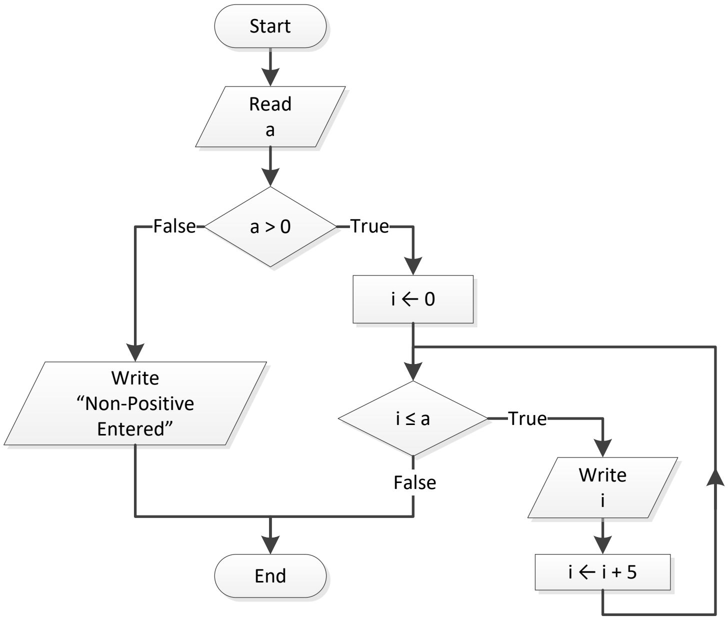
4. Solution



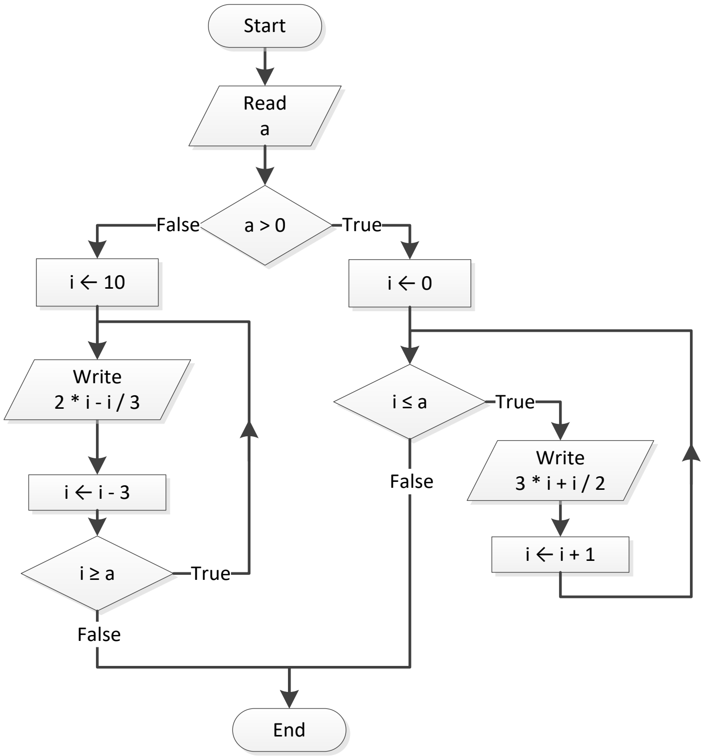
5. Solution



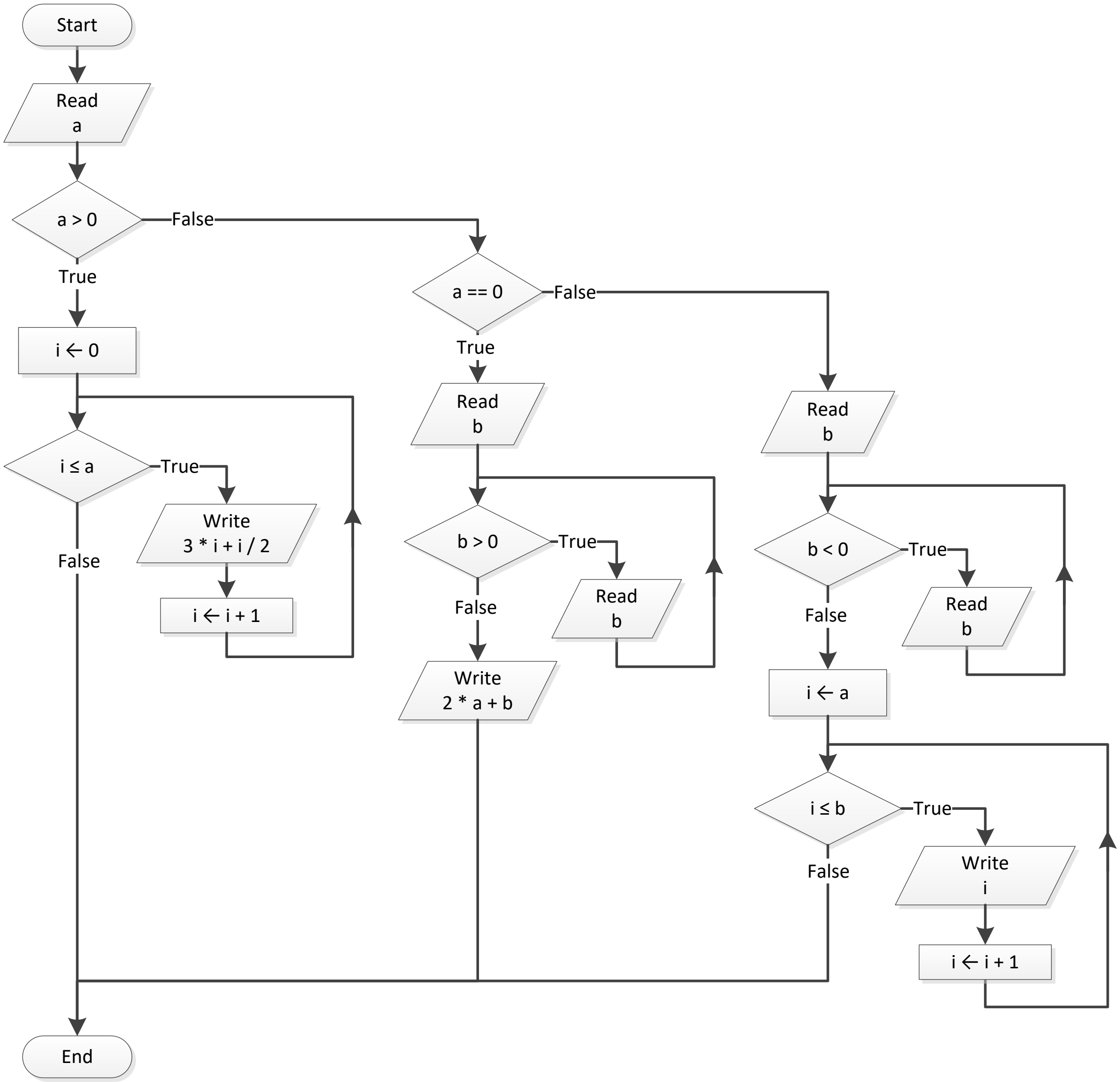
6. Solution



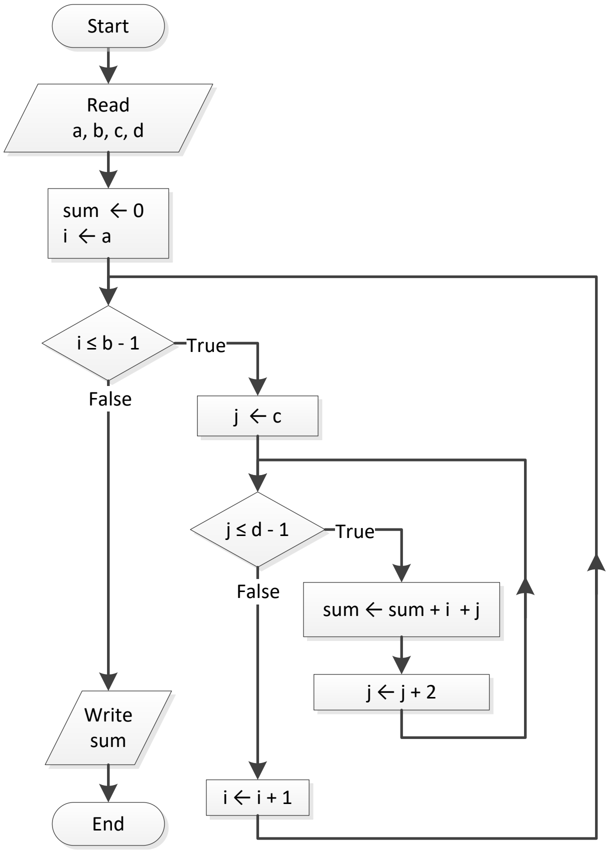
7. Solution


Start
Read
a
a > 0
False
True
i ← 10
i ← 0
Write
2 * i - i / 3
i ← i - 3
i ≥ a
True
False
i ≤ a
True
False
Write
3 * i + i / 2
i ← i + 1
End

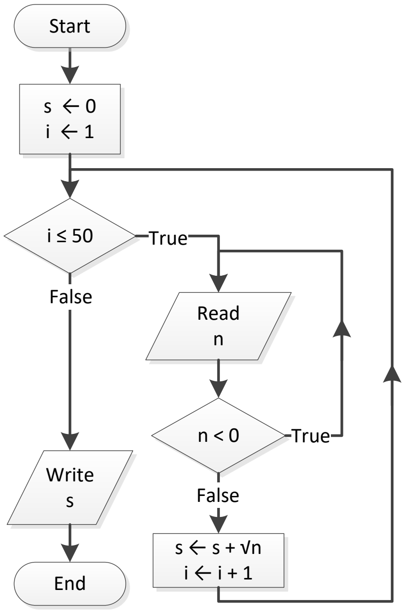
8. Solution


Start
Read a
a > 0
False
True
i ← 0
i ≤ a
True
Write 3 * i + i / 2
i ← i + 1
False
a == 0
False
True
Read b
b > 0
True
Read b
False
Write 2 * a + b
Read b
b < 0
True
Read b
False
i ← a
i ≤ b
True
Write i
i ← i + 1
False
End

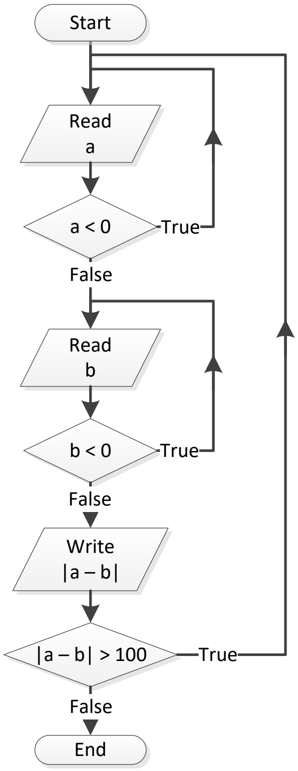
9. Solution



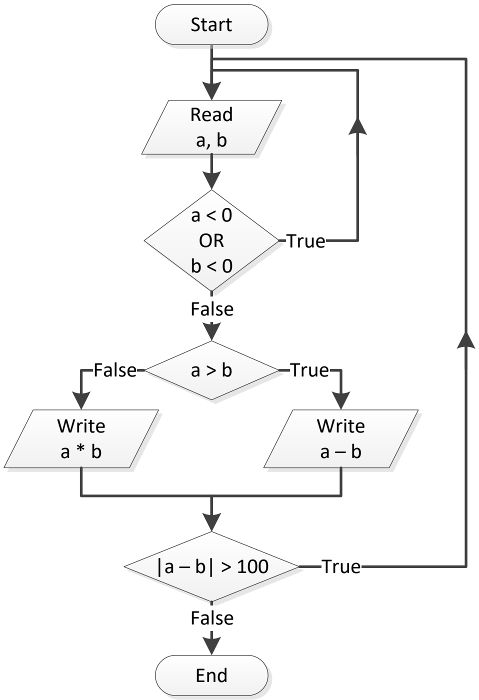
10. Solution



11. Solution



12. Solution



13. Solution

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

14. Solution

```
static void Main() {
  int a, i;

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

15. Solution

```
static void Main() {
  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());
  }
  Console.ReadKey();
}
```

16. Solution

```
static void Main() {
  int i;
  double P, S, a;

  i = 1;
  S = 0;
  P = 1;
  a = 0;
  if (i < 45) {
    S += a;
  }
  else {
    P *= a;
  }
  i++;
  while (i < 90) {
    a = Double.Parse(Console.ReadLine());
    if (i < 45) {
      S += a;
    }
    else {
      P *= a;
    }
    i++;
  }
  Console.Write(S + ", " + P);
  Console.ReadKey();
}
```

Chapter 31

31.6 Answers of Review Questions: True/False

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

31.7 Answers of Review Exercises

1. Solution

```
static void Main() {
  int i, s;

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

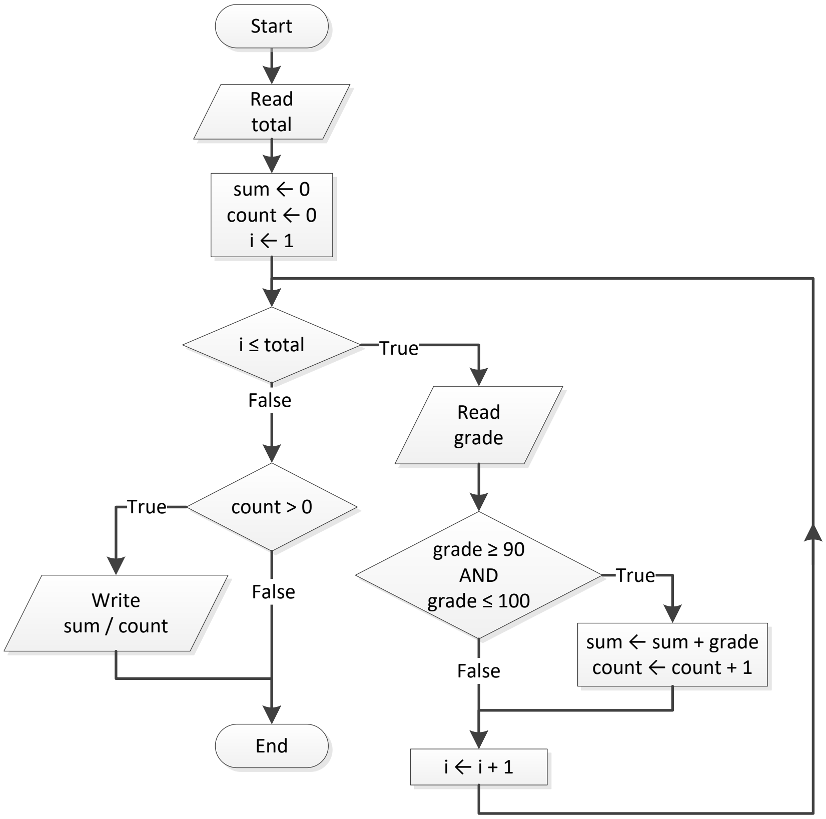
2. Solution

```
static void Main() {
  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.Write(p);
  Console.ReadKey();
}
```

3. Solution

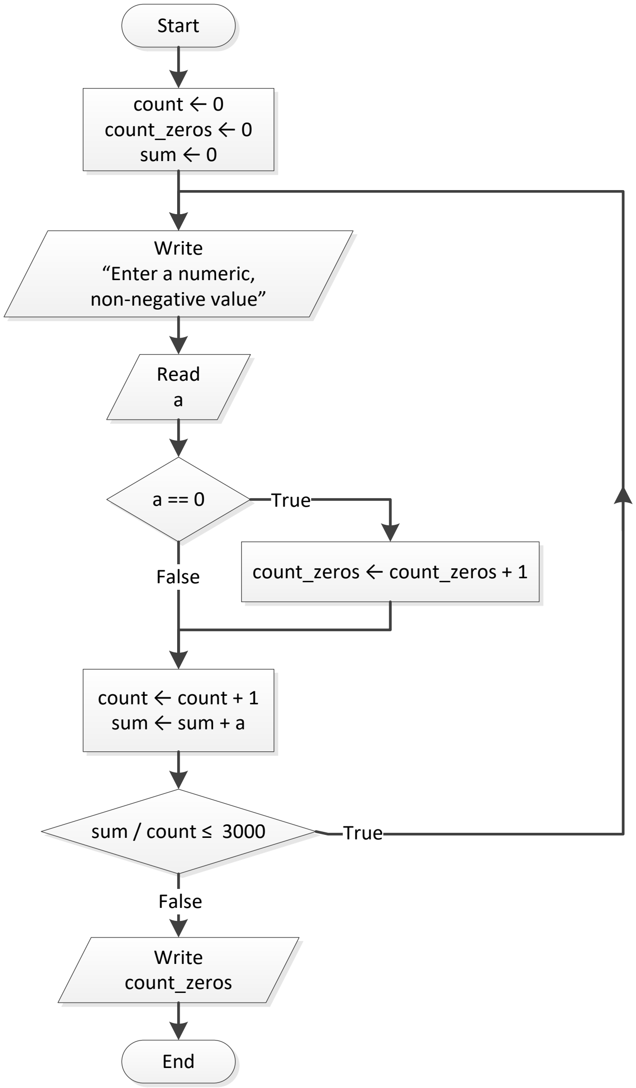
```
static void Main() {
  int i, offset, s;

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

4. Solution



```
static void Main() {
  int count, grade, i, sum, total;

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

5. Solution



```
static void Main() {
  int count, count_zeros;
  double a, sum;

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

6. Solution

First Approach

```
static void Main() {
  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);
    }
  }
  Console.ReadKey();
}
```

Second Approach

```
static void Main() {
  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);
          }
        }
      }
    }
  }
  Console.ReadKey();
}
```

7. Solution

First Approach

```
static void Main() {
  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);
    }
  }
  Console.ReadKey();
}
```

Second Approach

```
static void Main() {
  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);
          }
        }
      }
    }
  }
  Console.ReadKey();
}
```

8. Solution

```
input = Console.ReadLine();
while (Byte.TryParse(input, out x) == false || x != 1 && x != 0) {
    Console.WriteLine("Error");
    input = Console.ReadLine();
}
```

9. Solution

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

10. Solution

```
static void Main() {
  int count, x;
  double y;
  string input;

  Console.Write("Enter a non-negative number: ");
  input = Console.ReadLine();
  count = 1;
  while (Int32.TryParse(input, out x) == false || x < 0) {
    Console.WriteLine("Error: Invalid number!");
    Console.Write("Enter a non-negative number: ");
    input = Console.ReadLine();
    if (Int32.TryParse(input, out x) == false || x < 0) {
      count++;
      if (count == 3) break;
    }
  }

  if (count < 3) {
    y = Math.Sqrt(x);
    Console.WriteLine(y);
  }
  else {
    Console.WriteLine("Dude, you are dumb!");
  }
  Console.ReadKey();
}
```

11. Solution

```
static void Main() {
  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");
}
```

12. Solution

```
static void Main() {
  int a, b, c, d1, d2, d3, d4, r, x;
  string input;

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

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

  }
  Console.ReadKey();
}
```

13. Solution

```
static void Main() {
  int i;

  for (i = 0; i <= 30; i++) {
    Console.WriteLine(Math.Pow(2, i));
  }
  Console.ReadKey();
}
```

14. Solution

```
static void Main() {
  int i, offset;

  offset = 10;
  i = 1;
  while (i <= 401) {
    Console.WriteLine(i);
    i += offset;
    offset += 2;
  }
  Console.ReadKey();
}
```

15. Solution

```
static void Main() {
  int i;

  for (i = 1; i <= 100; i++) {
    Console.WriteLine(-i + "\n" + i);
  }
  Console.ReadKey();
}
```

16. Solution

First Approach

```
static void Main() {
  int i, offset, value;

  value = 0;
  for (i = 1; i <= 8; i++) {
    offset = Math.Pow(10, i - 1);
    value += offset;
    Console.WriteLine(value);
  }
  Console.ReadKey();
}
```

Second Approach

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

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

17. Solution

```
static void Main() {
  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;
  }
  Console.ReadKey();
}
```

18. Solution

```
static void Main() {
  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;
  }
  Console.ReadKey();
}
```

19. Solution

```
static void Main() {
  int denominator, i, n, nominator;
  double y;
  string input;

  Console.Write("Enter a positive integer: ");
  input = Console.ReadLine();
  while (Int32.TryParse(input, out n) == false || n < 1) {
    Console.Write("Wrong number. Please enter a positive integer: ");
    input = 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.Write(y);
  Console.ReadKey();
}
```

20. Solution

```
static void Main() {
  int i, n, nominator, sign;
  double y;
  string input;

  Console.Write("Enter a positive integer: ");
  input = Console.ReadLine();
  while (Int32.TryParse(input, out n) == false || n < 1) {
    Console.Write("Wrong number. Please enter a positive integer: ");
    input = 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.Write(y);
  Console.ReadKey();
}
```

21. Solution

```
static void Main() {
  int i, n, sign;
  double y;
  string input;

  Console.Write("Enter a positive integer: ");
  input = Console.ReadLine();
  while (Int32.TryParse(input, out n) == false || n < 1) {
    Console.Write("Wrong number. Please enter a positive integer: ");
    input = 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.Write(y);
  Console.ReadKey();
}
```

22. Solution

```
static void Main() {
  int i, n;
  double y;
  string input;

  Console.Write("Enter a positive integer: ");
  input = Console.ReadLine();
  while (Int32.TryParse(input, out n) == false || n < 1) {
    Console.Write("Wrong number. Please enter a positive integer: ");
    input = Console.ReadLine();
  }

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

  Console.Write(y);
  Console.ReadKey();
}
```

23. Solution

```
static void Main() {
  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.Write(factorial);
  Console.ReadKey();
}
```

Notice*: Please note that this C# code operates properly for all non-negative integers, including zero.*

24. Solution

```
static void Main() {
  int i;
  double max, sum, t;
  bool failure;
  string input;
  max = -460;
  sum = 0;
  for (i = 1; i <= 31; i++) {
    do {
      Console.Write("Enter temperature for day " + i + ": ");
      input = Console.ReadLine();

      failure = false;
      if (Double.TryParse(input, out t) == false) {
        Console.WriteLine("Please enter numeric values!");
        failure = true;
      }
      else if (t < -459.67) {
        Console.WriteLine("Please enter a value greater than 459.67");
        failure = true;
      }
    } while (failure == true);

    sum += t;
    if (t > max) {
      max = t;
    }
  }

  Console.Write(sum / 31 + ", " + max);
  Console.ReadKey();
}
```

25. Solution

```
static void Main() {
  int hour, max_hour, max_minutes, min_hour, min_minutes, minutes;
  double level, max, min;

  level = Double.Parse(Console.ReadLine());
  if (level != 9999) {
    hour = Int32.Parse(Console.ReadLine());
    minutes = Int32.Parse(Console.ReadLine());

    max = level;
    max_hour = hour;
    max_minutes = minutes;

    min = 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 > max) {
        max = level;
        max_hour = hour;
        max_minutes = minutes;
      }

      if (level < min) {
        min = level;
        min_hour = hour;
        min_minutes = minutes;
      }

      level = Double.Parse(Console.ReadLine());
    }

    Console.WriteLine(max + ", " + max_hour + ", " + max_minutes);
    Console.WriteLine(min + ", " + min_hour + ", " + min_minutes);
  }
  Console.ReadKey();
}
```

26. Solution

```
static void Main() {
  int a, b, c, i;
  string alphabet, input;
  bool failure;

  alphabet = "abcdefghijklmnopqrstuvwxyz";

  do {
    Console.Write("Enter an integer between 1 and 26: ");
    input = Console.ReadLine();

    failure = false;
    if (Int32.TryParse(input, out a) == false) {
      Console.WriteLine("Please enter numeric values!");
      failure = true;
    }
    else 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 == true);

  do {
    Console.Write("Enter an integer between 1 and 26: ");
    input = Console.ReadLine();

    failure = false;
    if (Int32.TryParse(input, out b) == false) {
      Console.WriteLine("Please enter numeric values!");
      failure = true;
    }
    else 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 == true);

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

  for (i = a; i <= b; i++) {
    Console.Write(alphabet[i - 1]);
  }
  Console.ReadKey();
}
```

27. Solution

```
static void Main() {
  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.Write("Attempts: " + attempts);
  Console.ReadKey();
}
```

28. Solution

```
static void Main() {
  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");
  }
  Console.ReadKey();
}
```

29. Solution

```
static void Main() {
  int choice, diagonal;

  do {
    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 == 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));
    }
  } while (choice != 3);
}
```

30. Solution

```
static void Main() {
  int count_a, count_a_boys, count_b, count_cdef_girls, grade;
  int i, max, min, n, sum, sum_a, sum_a_boys, sum_b;
  string sex, input;

  Console.Write("Enter total number of students: ");
  input = Console.ReadLine();
  while (Int32.TryParse(input, out n) == false || n < 1) {
    Console.Write("Wrong number. Please enter total number of students: ");
    input = Console.ReadLine();
  }

  sum = 0;
  sum_a = 0;
  count_a = 0;
  sum_b = 0;
  count_b = 0;
  sum_a_boys = 0;
  count_a_boys = 0;
  count_cdef_girls = 0;

  max = -1;
  min = 101;

  for (i = 1; i <= n; i++) {
    Console.Write("Enter grade for student No " + i + ": ");
    input = Console.ReadLine();
    while (Int32.TryParse(input, out grade) == false || grade < 0 || grade > 100) {
      Console.Write("Wrong grade. Please enter grade for student No " + i + ": ");
      input = Console.ReadLine();
    }

    Console.Write("Enter sex for student No " + i + ": ");
    sex = Console.ReadLine();
    sex = sex.ToUpper();
    while (sex != "M" && sex != "F") {
      Console.Write("Wrong sex. Please enter sex for student No " + i + ": ");
      sex = Console.ReadLine();
      sex = sex.ToUpper();
    }

    if (grade >= 90 && grade <= 100) {
      sum_a += grade;
      count_a++;
      if (sex == "M") {
        sum_a_boys += grade;
        count_a_boys++;
      }
    }
    else if (grade >= 80 && grade <= 89) {
      sum_b += grade;
      count_b++;
    }
    else {
      if (sex == "F") {
        count_cdef_girls++;
      }
    }

    if (grade > max) {
      max = grade;
    }

    if (grade < min) {
      min = grade;
    }

    sum += grade;
  }

  if (count_a > 0) {
    Console.Write("The average value of those who got an 'A' is: ");
    Console.WriteLine(sum_a / (double)count_a);
  }
  if (count_b > 0) {
    Console.Write("The average value of those who got a 'B' is: ");
    Console.WriteLine(sum_b / (double)count_b);
  }
  if (count_a_boys > 0) {
    Console.Write("The average value of boys who got an 'A' is: ");
    Console.WriteLine(sum_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: " + max);
  Console.WriteLine("The lowest grade is: " + min);
  Console.Write("The average grade of the whole class is: " + sum / (double)n);
  Console.ReadKey();
}
```

31. Solution

```
static void Main() {
  double amount, discount;
  string answer, input;

  do {
    Console.Write("Enter amount: ");
    input = Console.ReadLine();
    while (Double.TryParse(input, out amount) == false || amount < 0) {
      Console.WriteLine("Wrong amount. Please enter amount: ");
      input = 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();
  } while (answer.ToUpper() == "YES");
}
```

32. Solution

```
const double TAX_RATE = 0.25;

static void Main() {
  int kwh;
  double t;
  string input;

  Console.Write("Enter number of Kilowatt-hours consumed: ");
  input = Console.ReadLine();
  while (Int32.TryParse(input, out kwh) == false || kwh < 0 && kwh != -1) {
    Console.Write("Wrong value. Please enter number of Kilowatt-hours consumed: ");
    input = 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: ");
    input = Console.ReadLine());
    while (Int32.TryParse(input, out kwh) == false || kwh < 0 && kwh != -1) {
      Console.Write("Wrong value. Please enter number of Kilowatt-hours consumed: ");
      input = Console.ReadLine();
    }
  }
  Console.ReadKey();
}
```

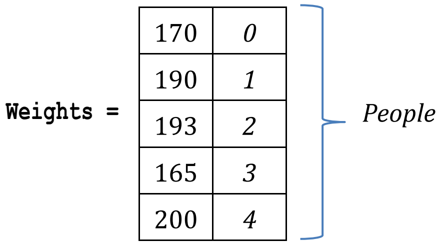
Chapter 32

32.3 Answers of Review Questions: True/False

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

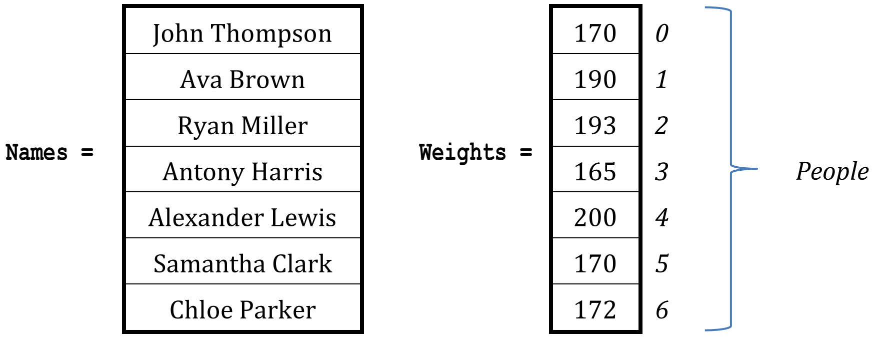
32.4 Answers of Review Exercises

1. Solution



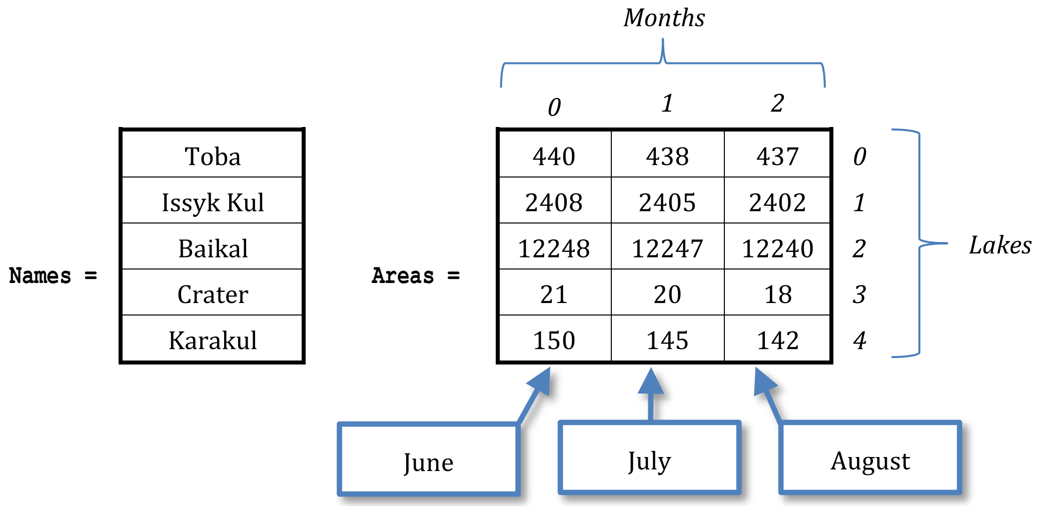
170	0
190	1
193	2
165	3
200	4

2. Solution



Names	Weights	
John Thompson	170	0
Ava Brown	190	1
Ryan Miller	193	2
Antony Harris	165	3
Alexander Lewis	200	4
Samantha Clark	170	5
Chloe Parker	172	6

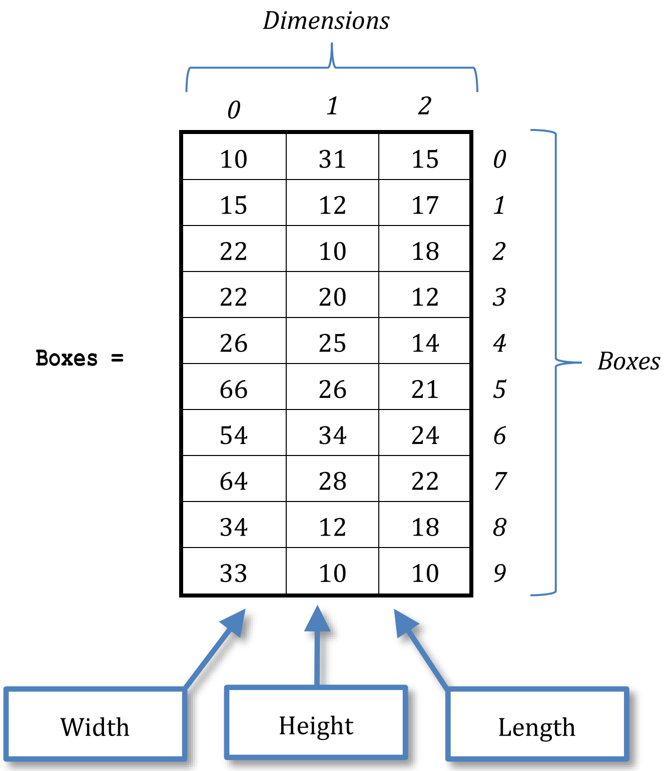
3. Solution



Names
Toba
Issyk Kul
Baikal
Crater
Karakul

0	1	2	
440	438	437	0
2408	2405	2402	1
12248	12247	12240	2
21	20	18	3
150	145	142	4

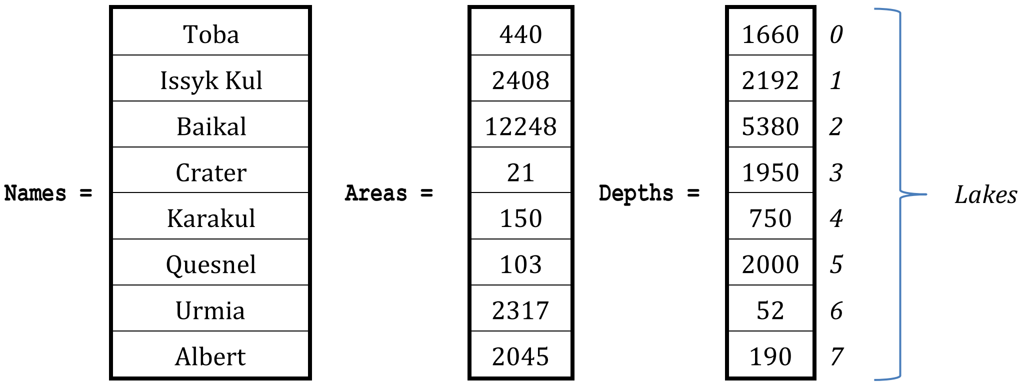
4. Solution



`Boxes =`

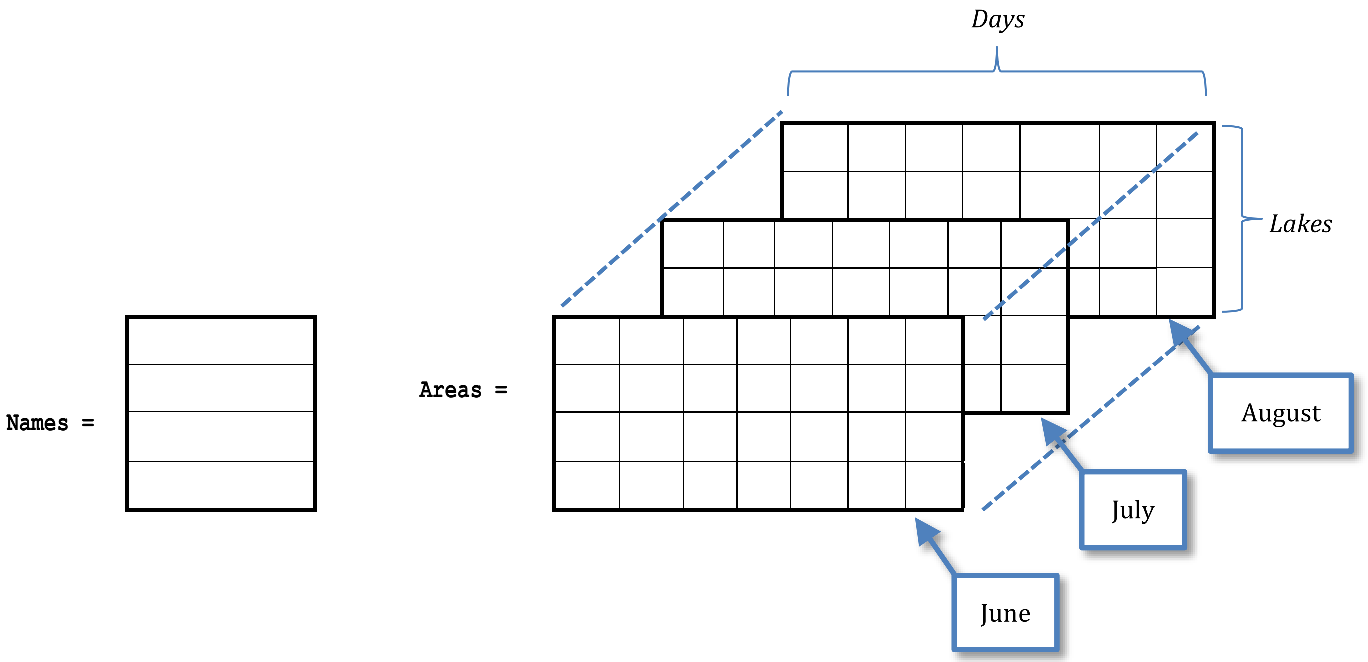
0	1	2	
10	31	15	0
15	12	17	1
22	10	18	2
22	20	12	3
26	25	14	4
66	26	21	5
54	34	24	6
64	28	22	7
34	12	18	8
33	10	10	9

5. Solution



`Names =`	`Areas =`	`Depths =`	
Toba	440	1660	0
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



Chapter 33

33.5 Answers of Review Questions: True/False

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

33.6 Answers of Review Questions: Multiple Choice

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

33.7 Answers of Review Exercises

1. 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

2. 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

3. *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	`a[x] = 3`	0	**3**	1	?	?
5	`a[a[0]] = a[x + 1] % 2`	0	3	1	?	**3**
6	`a[a[0] % 2] = 10`	0	3	**10**	?	3
7	`x++`	**1**	3	10	?	3
8	`a[x + 1] = a[x] + 9`	1	3	10	**19**	3

4. Solution

For input value of 100

Step	Statement	x	a[0]	a[1]	a[2]	a[3]
1	`int[] a = new int[4]`	?	?	?	?	?
2	`a[1] = Int32.Parse(Console.ReadLine())`	?	?	**100**	?	?
3	`x = 0`	**0**	?	100	?	?
4	`a[x] = 3`	0	**3**	100	?	?
5	`a[a[0]] = a[x + 1] % 10`	0	3	100	?	**0**
6	`if (a[3] > 5)`	False				
7	`a[2] = 3`	0	3	100	**3**	0

For input value of 108

Step	Statement	x	a[0]	a[1]	a[2]	a[3]
1	`int[] a = new int[4]`	?	?	?	?	?
2	`a[1] = Int32.Parse(Console.ReadLine())`	?	?	**108**	?	?
3	`x = 0`	**0**	?	108	?	?
4	`a[x] = 3`	0	**3**	108	?	?
5	`a[a[0]] = a[x + 1] % 10`	0	3	108	?	**8**
6	`if (a[3] > 5)`	True				
7	`a[a[0] % 2] = 9`	0	3	**9**	?	8
8	`x += 1`	**1**	3	9	?	8
9	`a[x + 1] = a[x] + 9`	1	3	9	**18**	8

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	`a[x] = 3`	0	**3**	1	?	?
5	`a[a[0]] = a[x + 1] % 10`	0	3	1	?	**1**
6	`if (a[3] > 5)`	False				
7	`a[2] = 3`	0	3	1	**3**	1

5. 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)`	4	3	**1**	?	?

	a[0] = 1;
	else
	  a[0] = y;					

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**

6. 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	11	**50**
26	i++	**6**	18	11	46	11	11	50
27	i <= 5	False						

7. 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						

8. Solution

```
const int ELEMENTS = 100;

static void Main() {
  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));
  }
  Console.ReadKey();
}
```

9. Solution

```
const int ELEMENTS = 80;

static void Main() {
  int i;

  double[] a = new double[ELEMENTS];
  for (i = 0; i <= ELEMENTS - 1; i++) {
    a[i] = Double.Parse(Console.ReadLine());
  }
  for (i = ELEMENTS - 1; i >= 0; i--) {
    Console.WriteLine(Math.Pow(a[i], 2));
  }
  Console.ReadKey();
}
```

10. Solution

```
const int ELEMENTS = 90;

static void Main() {
  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]);
    }
  }
  Console.ReadKey();
}
```

11. Solution

```
const int ELEMENTS = 50;

static void Main() {
  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]);
    }
  }
  Console.ReadKey();
}
```

12. Solution

```
const int ELEMENTS = 30;

static void Main() {
  int i;
  double sum;

  double[] a = new double[ELEMENTS];
  for (i = 0; i <= ELEMENTS - 1; i++) {
    a[i] = Double.Parse(Console.ReadLine());
  }

  sum = 0;
  for (i = 0; i <= ELEMENTS - 1; i++) {
    if (a[i] > 0) {
      sum += a[i];
    }
  }
  Console.Write(sum);
  Console.ReadKey();
}
```

13. Solution

```
const int ELEMENTS = 50;

static void Main() {
  int i, sum;

  int[] a = new int[ELEMENTS];
  for (i = 0; i <= ELEMENTS - 1; i++) {
    a[i] = Int32.Parse(Console.ReadLine());
  }

  sum = 0;
  for (i = 0; i <= ELEMENTS - 1; i++) {
    if (a[i] >= 10 && a[i] <= 99) {
      sum += a[i];
    }
  }
  Console.Write(sum);
  Console.ReadKey();
}
```

14. Solution

```
const int ELEMENTS = 40;

static void Main() {
  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.Write(sum_pos + ", " + sum_neg);
  Console.ReadKey();
}
```

15. Solution

```
const int ELEMENTS = 20;

static void Main() {
  int i;
  double sum;

  double[] a = new double[ELEMENTS];
  for (i = 0; i <= ELEMENTS - 1; i++) {
    a[i] = Double.Parse(Console.ReadLine());
  }

  sum = 0;
  for (i = 0; i <= ELEMENTS - 1; i++) {
    sum += a[i];
  }
  Console.Write(sum / ELEMENTS);
  Console.ReadKey();
}
```

16. Solution

```
const int WORDS = 50;

static void Main() {
  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]);
    }
  }
  Console.ReadKey();
}
```

17. Solution

```
const int WORDS = 40;

static void Main() {
  int count, i, j;

  string[] a = new string[WORDS];
  for (i = 0; i <= WORDS - 1; i++) {
    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") {
        count++;
      }
    }
    if (count >= 2) {
      Console.WriteLine(a[i]);
    }
  }
  Console.ReadKey();
}
```

Chapter 34

34.7 Answers of Review Questions: True/False

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

34.8 Answers of Review Questions: Multiple Choice

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

34.9 Answers of Review Exercises

1. Solution

Step	Statement	x	a
1	`int[,] a = new int[2, 3]`	?	? ? ? ? ? ?
2	`a[0, 2] = 1`	?	? ? **1** ? ? ?
3	`x = 0`	**0**	? ? 1 ? ? ?
4	`a[0, x] = 9`	0	**9** ? 1 ? ? ?

5	`a[0, x + a[0, 2]] = 4`	0	9 **4** 1 ? ? ?
6	`a[a[0, 2], 2] = 19`	0	9 4 1 ? ? **19**
7	`a[a[0, 2], x + 1] = 13`	0	9 4 1 ? **13** 19
8	`a[a[0, 2], x] = 15`	0	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	**5** ? ? ? ? ?
7	`j++`	0	**1**	5 ? ? ? ? ?
8	`j <= 2`	True		
9	`a[i, j] = (i + 1) * 5 + j`	0	1	5 **6** ? ? ? ?
10	`j++`	0	**2**	5 6 ? ? ? ?
11	`j <= 2`	True		

12	`a[i, j] = (i + 1) * 5 + j`	0	2	5 6 **7** ? ? ?
13	`j++`	0	**3**	5 6 7 ? ? ?
14	`j <= 2`	False		
15	`i++`	**1**	3	5 6 7 ? ? ?
16	`i <= 1`	True		
17	`j = 0`	1	**0**	5 6 7 ? ? ?
18	`j <= 2`	True		
19	`a[i, j] = (i + 1) * 5 + j`	1	0	5 6 7 **10** ? ?
20	`j++`	1	**1**	5 6 7 10 ? ?
21	`j <= 2`	True		
22	`a[i, j] = (i + 1) * 5 + j`	1	1	5 6 7 10 **11** ?
23	`j++`	1	**2**	5 6 ? 10 11 ?
24	`j <= 2`	True		
25	`a[i, j] = (i + 1) * 5 + j`	1	2	5 6 ? 10 11 **12**
26	`j++`	1	**3**	5 6 7 10 11 12
27	`j <= 2`	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** ? ? 6 ? ?
10	`i++`	**2**	0	2 ? ? 4 ? ? 6 ? ?
11	`i <= 2`	True		
12	`a[i, j] = (i + 1) * 2 + j * 4`	2	0	2 ? ? 4 ? ? **6** ? ?
13	`i++`	**3**	0	2 ? ? 4 ? ? 6 ? ?
14	`i <= 2`	False		

15	`j++`	3	**1**	2 ? ? 4 ? ? 6 ? ?
16	`j <= 2`	True		
17	`i = 0`	**0**	1	2 ? ? 4 ? ? 6 ? ?
18	`i <= 2`	True		
19	`a[i, j] = (i + 1) * 2 + j * 4`	0	1	2 **6** ? 4 ? ? 6 ? ?
20	`i++`	**1**	1	2 6 ? 4 ? ? 6 ? ?
21	`i <= 2`	True		
22	`a[i, j] = (i + 1) * 2 + j * 4`	1	1	2 6 ? 4 **8** ? 6 ? ?
23	`i++`	**2**	1	2 6 ? 4 8 ? 6 ? ?
24	`i <= 2`	True		
25	`a[i, j] = (i + 1) * 2 + j * 4`	2	1	2 6 ? 4 8 ? 6 **10** ?
26	`i++`	**3**	1	2 6 ? 4 8 ? 6 10 ?
27	`i <= 2`	False		
28	`j++`	3	**2**	2 6 ? 4 8 ? 6 10 ?

29	`j <= 2`	True		
30	`i = 0`	**0**	2	2 6 ? 4 8 ? 6 10 ?
31	`i <= 2`	True		
32	`a[i, j] = (i + 1) * 2 + j * 4`	0	2	2 6 **10** 4 8 ? 6 10 ?
33	`i++`	**1**	2	2 6 10 4 8 ? 6 10 ?
34	`i <= 2`	True		
35	`a[i, j] = (i + 1) * 2 + j * 4`	1	2	2 6 10 4 8 **12** 6 10 ?
36	`i++`	**2**	2	2 6 10 4 8 12 6 10 ?
37	`i <= 2`	True		
38	`a[i, j] = (i + 1) * 2 + j * 4`	2	2	2 6 10 4 8 12 6 10 **14**
39	`i++`	**3**	2	2 6 10 4 8 12 6 10 14
40	`i <= 2`	False		
41	`j++`	3	**3**	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() {
  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);
      }
    }
  }
  Console.ReadKey();
}
```

10. Solution

```
const int ROWS = 10;
const int COLUMNS = 6;

static void Main() {
  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]);
    }
  }
  Console.ReadKey();
}
```

11. Solution

```
const int ROWS = 12;
const int COLUMNS = 8;
```

```
static void Main() {
  int i, j;
  double sum;

  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());
    }
  }

  sum = 0;
  for (i = 1; i <= ROWS - 1; i += 2) {
    for (j = 0; j <= COLUMNS - 1; j += 2) {
      sum += a[i, j];
    }
  }
  Console.Write(sum);
  Console.ReadKey();
}
```

12. Solution

```
const int N = 8 ;

static void Main() {
  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.Write(sum_diagonal / N + ", " + sum_antidiagonal / N);
  Console.ReadKey();
}
```

13. Solution

```
const int N = 5;

static void Main() {
  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();
  }
  Console.ReadKey();
}
```

14. Solution

```
const int N = 5;

static void Main() {
  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();
  }
  Console.ReadKey();
}
```

15. Solution

```
const int ROWS = 5;
const int COLUMNS = 4;

static void Main() {
  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);
      }
    }
  }
  Console.ReadKey();
}
```

16. Solution

```
const int ROWS = 10;
const int COLUMNS = 4;

static void Main() {
  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.Write(count);
  Console.ReadKey();
}
```

17. Solution

```
const int ROWS = 3;
const int COLUMNS = 4;

static void Main() {
  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] + " ");
    }
  }
  Console.ReadKey();
}
```

18. Solution

```
const int ROWS = 20;
const int COLUMNS = 14;

static void Main() {
  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]);
      }
    }
  }
  Console.ReadKey();
}
```

19. Solution

First Approach

```
const int ROWS = 20;
const int COLUMNS = 14;

static void Main() {
  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]);
        }
      }
    }
  }
  Console.ReadKey();
}
```

Second Approach

```
const int ROWS = 20;
const int COLUMNS = 14;

static void Main() {
  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]);
        }
      }
    }
  }
  Console.ReadKey();
}
```

Chapter 35

35.7 Answers of Review Questions: True/False

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

35.8 Answers of Review Questions: Multiple Choice

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

35.9 Answers of Review Exercises

1. Solution

```
const int STUDENTS = 15;
const int TESTS = 5;

static void Main() {
  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.Write("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");
    }
  }
  Console.ReadKey();
}
```

2. Solution

```
const int OBJECTS = 5;
const int FALLS = 10;

static void Main() {
  int i, j, sum;

  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++) {
    sum = 0;
    for (j = 0; j <= FALLS - 1; j++) {
      sum += g[i, j];
    }
    Console.WriteLine("Average g for object No " + (i + 1) + ": " + (sum / (double)FALLS));
  }

  for (j = 0; j <= FALLS - 1; j++) {
    sum = 0;
    for (i = 0; i <= OBJECTS - 1; i++) {
      sum += g[i, j];
    }
    Console.WriteLine("Average g for fall No " + (j + 1) + ": " + (sum / (double)OBJECTS));
  }

  sum = 0;
  for (i = 0; i <= OBJECTS - 1; i++) {
    for (j = 0; j <= FALLS - 1; j++) {
      sum += g[i, j];
    }
  }
  Console.Write("Overall average g: " + (sum / (double)(OBJECTS * FALLS)));
  Console.ReadKey();
}
```

3. Solution

```
const int PLAYERS = 15;
const int MATCHES = 12;

static void Main() {
  int i, j, sum;

  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++) {
    sum = 0;
    for (j = 0; j <= MATCHES - 1; j++) {
      sum += points[i, j];
    }
    Console.WriteLine("Total number of points for player No " + (i + 1) + ": " + sum);
  }

  for (j = 0; j <= MATCHES - 1; j++) {
    sum = 0;
    for (i = 0; i <= PLAYERS - 1; i++) {
      sum += points[i, j];
    }
    Console.WriteLine("Total number of points for match No " + (j + 1) + ": " + sum);
  }
  Console.ReadKey();
}
```

4. Solution

```
const int CITIES = 20;
const int HOURS = 24;

static void Main() {
  int i, j;
  double sum;

  double[,] temperatures = new double[CITIES, HOURS];
  for (i = 0; i <= CITIES - 1; i++) {
    for (j = 0; j <= HOURS - 1; j++) {
      temperatures[i, j] = Double.Parse(Console.ReadLine());
    }
  }

  for (j = 0; j <= HOURS - 1; j++) {
    sum = 0;
    for (i = 0; i <= CITIES - 1; i++) {
      sum += temperatures[i, j];
    }
    if (sum / CITIES < 10) {
      Console.WriteLine("Hour: " + (j + 1));
    }
  }
  Console.ReadKey();
}
```

5. Solution

```
const int PLAYERS = 24;
const int MATCHES = 10;

static void Main() {
  int i, j, sum;

  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++) {
    sum = 0;
    for (j = 0; j <= MATCHES - 1; j++) {
      sum += goals[i, j];
    }
    Console.WriteLine(names[i] + ": " + (sum / (double)MATCHES));
  }

  for (j = 0; j <= MATCHES - 1; j++) {
    sum = 0;
    for (i = 0; i <= PLAYERS - 1; i++) {
      sum += goals[i, j];
    }
    Console.WriteLine("Match No " + (j + 1) + ": " + sum);
  }
  Console.ReadKey();
}
```

6. Solution

```
const int STUDENTS = 24;
const int LESSONS = 10;

static void Main() {
  int i, j, sum;

  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++) {
    sum = 0;
    for (j = 0; j <= LESSONS - 1; j++) {
      sum += grades[i, j];
    }
    average[i] = sum / (double)LESSONS;
    Console.WriteLine(names[i] + ": " + average[i]);
  }

  for (j = 0; j <= LESSONS - 1; j++) {
    sum = 0;
    for (i = 0; i <= STUDENTS - 1; i++) {
      sum += grades[i, j];
    }
    Console.WriteLine(sum / (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!");
    }
  }
  Console.ReadKey();
}
```

7. Solution

```
const int ARTISTS = 15;
const int JUDGES = 5;

static void Main() {
  int i, j, sum;

  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++) {
    sum = 0;
    for (j = 0; j <= JUDGES - 1; j++) {
      sum += score[i, j];
    }
    Console.WriteLine(artist_names[i] + ", " + song_titles[i] + ": " + sum);
  }

  for (j = 0; j <= JUDGES - 1; j++) {
    sum = 0;
    for (i = 0; i <= ARTISTS - 1; i++) {
      sum += score[i, j];
    }
    Console.WriteLine(judge_names[j] + ": " + sum / (double)ARTISTS);
  }
  Console.ReadKey();
}
```

8. Solution

```
const int PEOPLE = 30;
const int MONTHS = 12;

static void Main() {
  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));
  }
  Console.ReadKey();
}
```

9. Solution

```
const double VAT = 0.19;
const int CONSUMERS = 1000;

static void Main() {
  int consumed, i;
  double payment, sum;


  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());
  }

  sum = 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);

    sum += consumed;
  }

  Console.Write(sum + ", " + (sum * 0.07 + sum * 0.07 * VAT));
  Console.ReadKey();
}
```

10. Solution

```
const int CURRENCIES = 4;
const int DAYS = 5;

static void Main() {
  int i, j;
  double average, sum, usd;

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

  string[] currency = {"British Pound Sterling", "Euro", "Canadian Dollar", "Australian Dollar"};

  double[,] rate = { {1.579, 1.577, 1.572, 1.580, 1.584},
                     {1.269, 1.270, 1.265, 1.240, 1.255},
                     {0.895, 0.899, 0.884, 0.888, 0.863},
                     {0.811, 0.815, 0.822, 0.829, 0.819}
                   };

  for (i = 0; i <= CURRENCIES - 1; i++) {
    sum = 0;
    for (j = 0; j <= DAYS - 1; j++) {
      sum += rate[i, j];
    }
    average = sum / DAYS;
    Console.WriteLine(usd + " US dollars = " + (usd / average) + " " + currency[i] + "s");
  }
  Console.ReadKey();
}
```

11. Solution

```
const int EMPLOYEES = 10;
const int DAYS = 5;

static void Main() {
  int i, j;
  double gross_pay, pay_rate, sum;

  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++) {
    sum = 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);
      }
      sum += gross_pay;
    }
    Console.WriteLine(days[j] + ", " + sum);
  }
  Console.ReadKey();
}
```

12. Solution

```
const int ROWS = 3;
const int COLUMNS = 4;
const int ELEMENTS = ROWS * COLUMNS;

static void Main() {
  int i, j, k;

  int[,] a = { {9, 9, 2, 6},
               {4, 1, 10, 11},
               {12, 15, 7, 3}
             };

  int[] b = new int[ELEMENTS];
  k = 0;
  for (i = 0; i <= ROWS - 1; i++) {
    for (j = 0; j <= COLUMNS - 1; j++) {
      b[k] = a[i, j];
      k++;
    }
  }

  for (k = 0; k <= ELEMENTS - 1; k++) {
    Console.Write(b[k] + " ");
  }
  Console.ReadKey();
}
```

13. Solution

```
const int ROWS = 3;
const int COLUMNS = 3;

static void Main() {
  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];
      k++;
    }
  }

  for (i = 0; i <= ROWS - 1; i++) {
    for (j = 0; j <= COLUMNS - 1; j++) {
      Console.Write(b[i, j] + "\t");
    }
    Console.WriteLine();
  }
  Console.ReadKey();
}
```

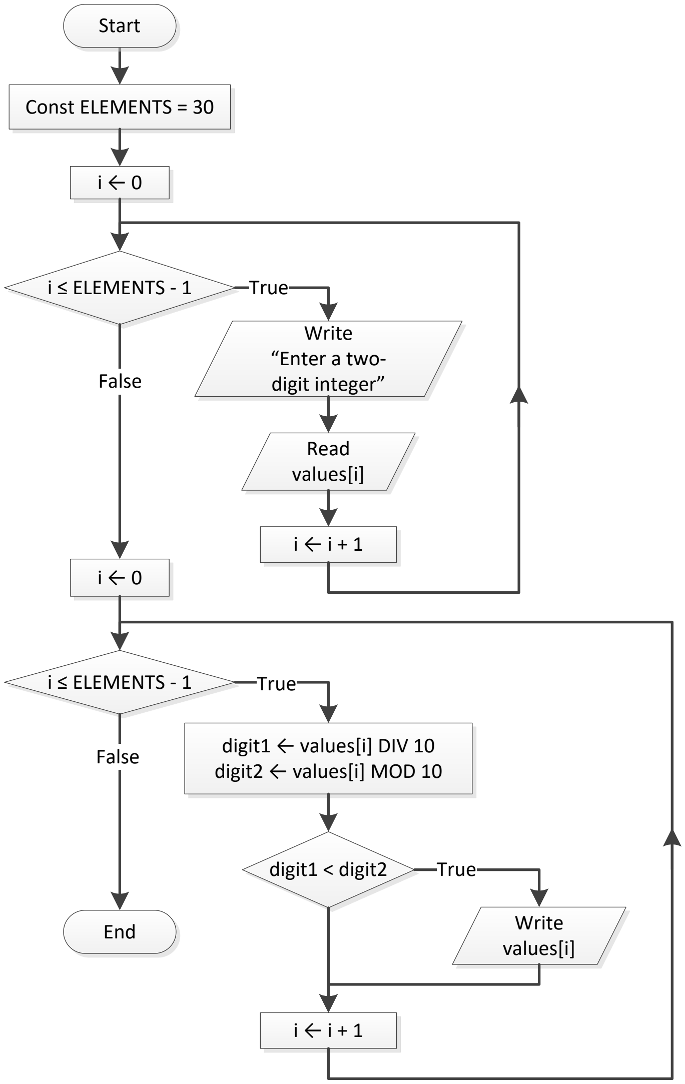
Chapter 36

36.7 Answers of Review Questions: True/False

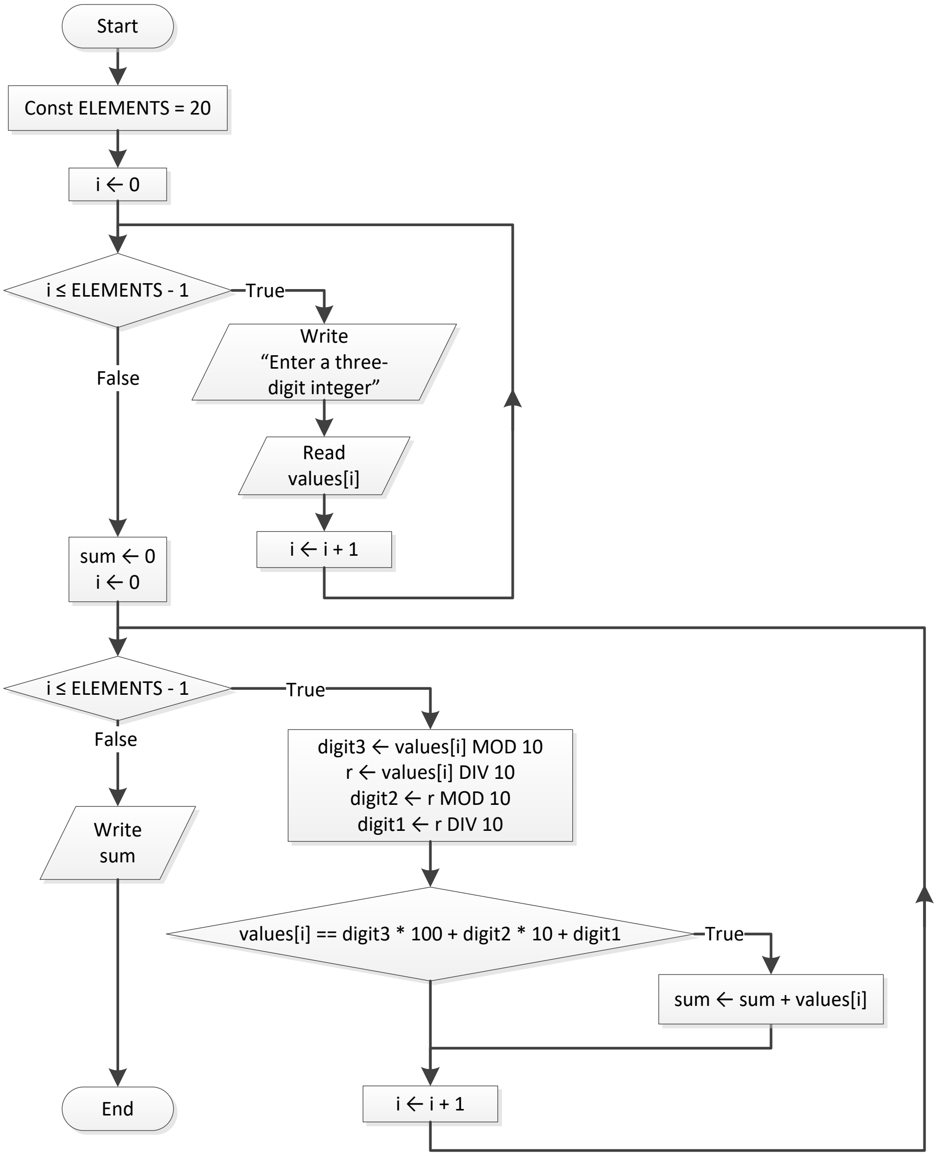
1. true
2. false
3. true
4. true
5. true
6. true
7. true
8. true
9. false
10. false
11. false
12. true
13. false
14. true
15. false
16. true
17. false
18. true
19. true
20. true
21. false
22. true
23. true
24. true
25. true
26. false
27. true
28. true
29. true

36.8 Answers of Review Exercises

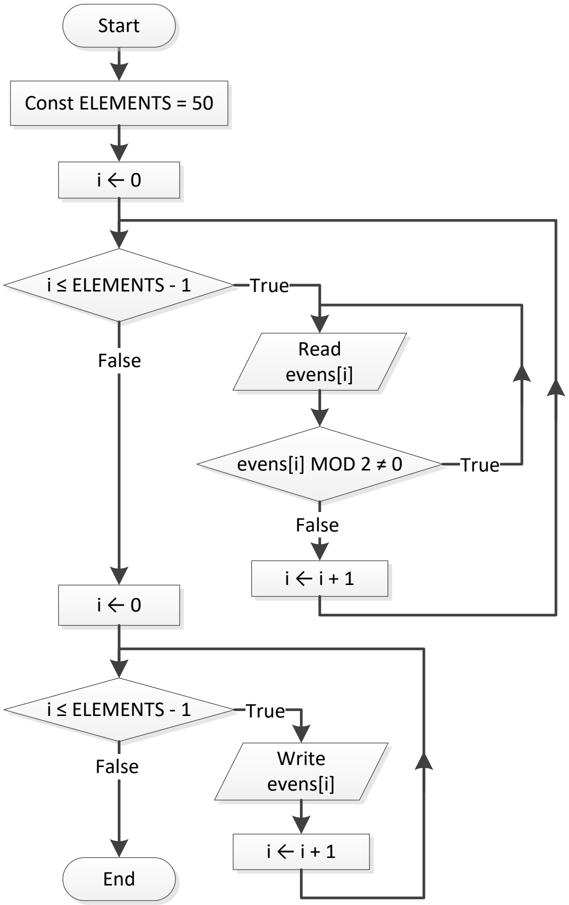
1. Solution



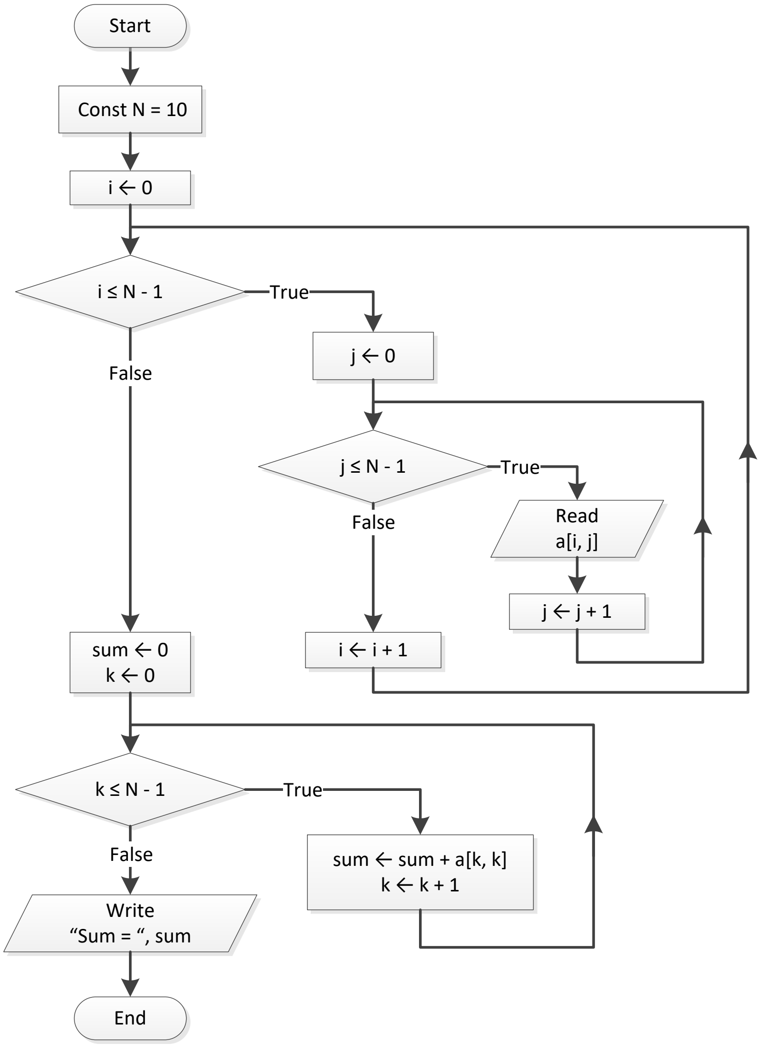
2. Solution


Start
Const ELEMENTS = 20
i ← 0
i ≤ ELEMENTS - 1
True
False
Write
"Enter a three-digit integer"
Read
values[i]
i ← i + 1
sum ← 0
i ← 0
i ≤ ELEMENTS - 1
True
False
digit3 ← values[i] MOD 10
r ← values[i] DIV 10
digit2 ← r MOD 10
digit1 ← r DIV 10
Write
sum
values[i] == digit3 * 100 + digit2 * 10 + digit1
True
sum ← sum + values[i]
i ← i + 1
End

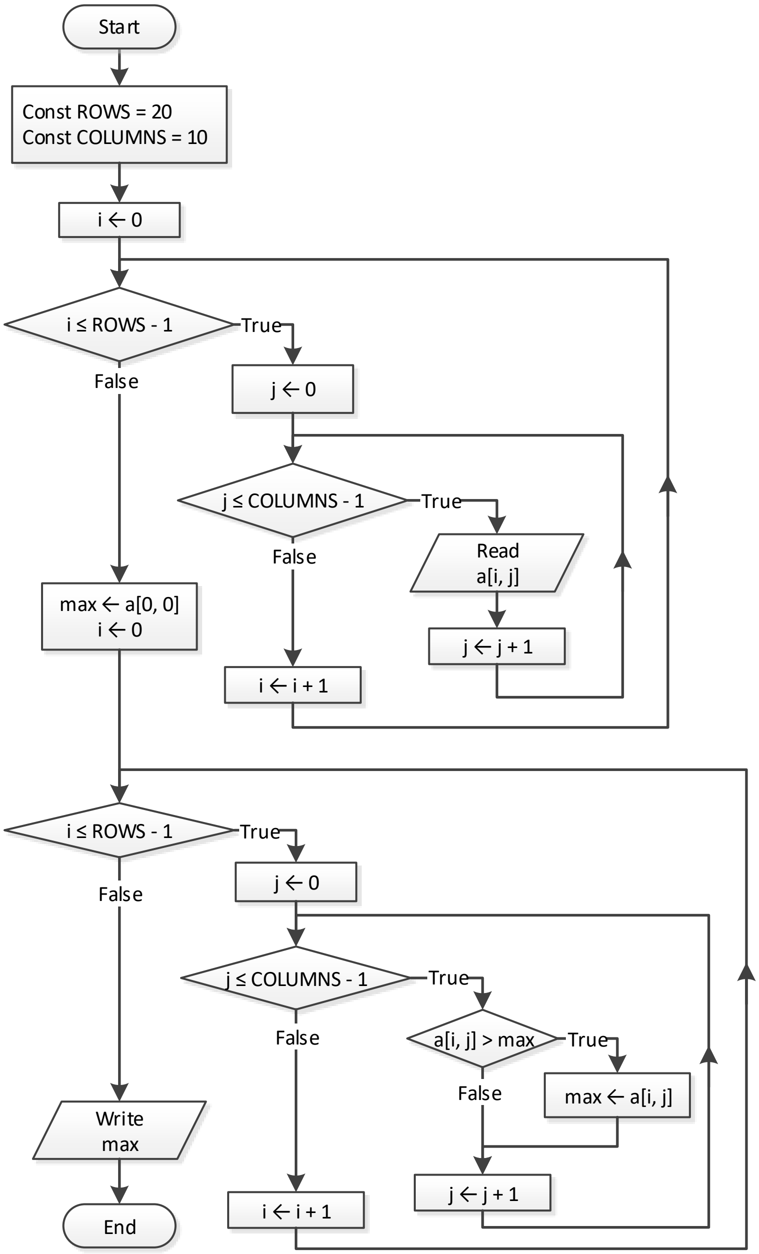
3. Solution


Start
Const ELEMENTS = 50
i ← 0
i ≤ ELEMENTS - 1
True
False
Read evens[i]
evens[i] MOD 2 ≠ 0
True
False
i ← i + 1
i ← 0
i ≤ ELEMENTS - 1
True
False
Write evens[i]
i ← i + 1
End

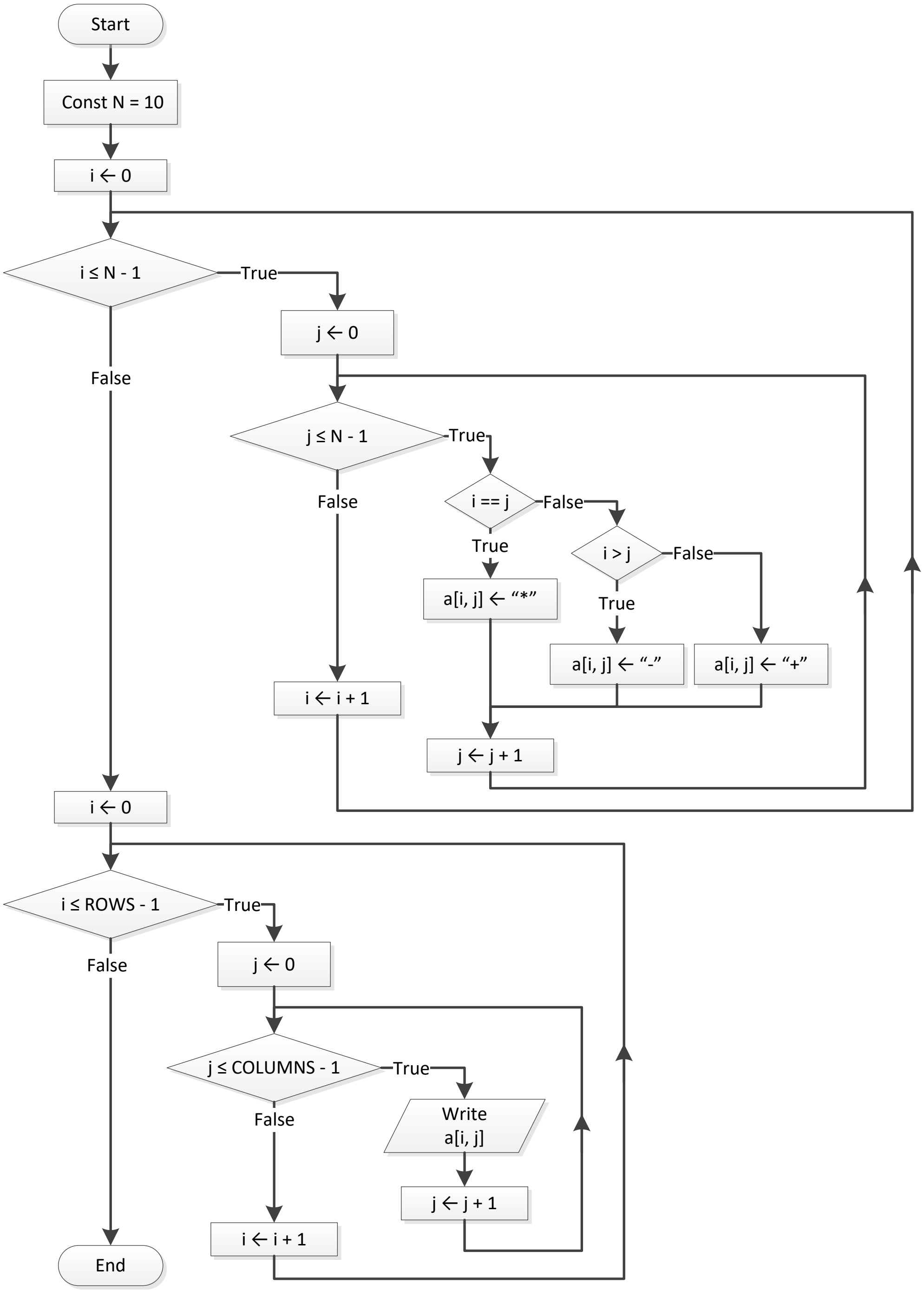
4. Solution


Start
Const N = 10
i ← 0
i ≤ N - 1
True
False
j ← 0
j ≤ N - 1
True
False
Read
a[i, j]
j ← j + 1
i ← i + 1
sum ← 0
k ← 0
k ≤ N - 1
True
False
sum ← sum + a[k, k]
k ← k + 1
Write
“Sum = “, sum
End

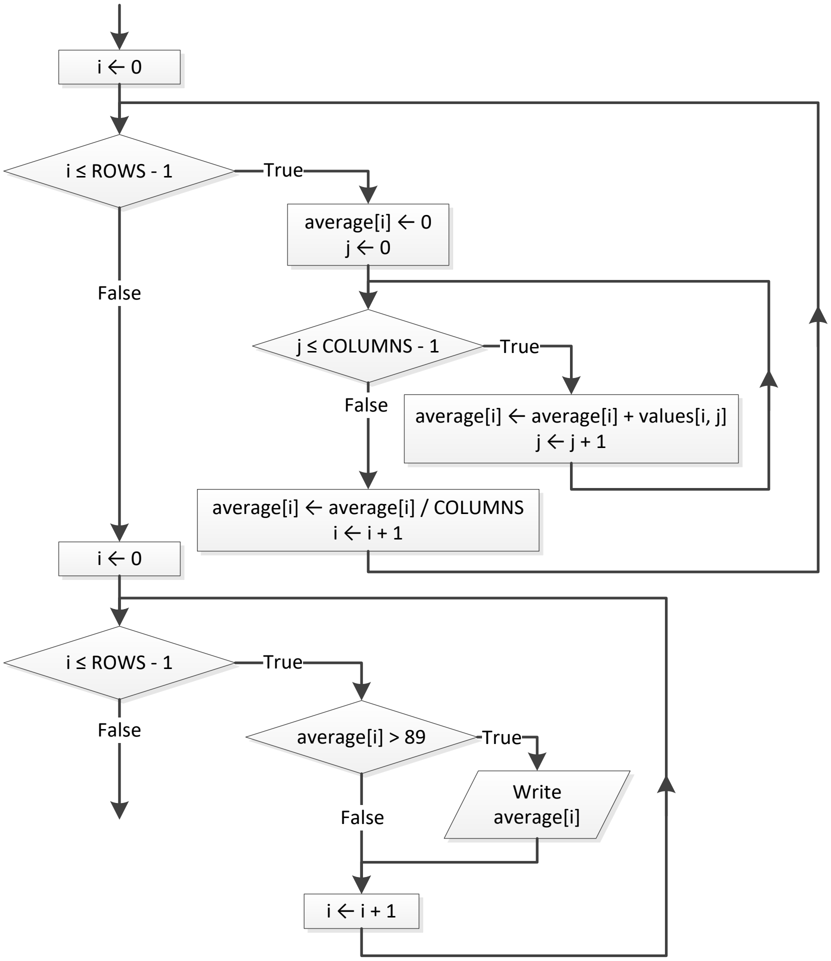
5. Solution


Start
Const ROWS = 20
Const COLUMNS = 10
i ← 0
i ≤ ROWS - 1
True
False
j ← 0
j ≤ COLUMNS - 1
True
False
Read
a[i, j]
j ← j + 1
i ← i + 1
max ← a[0, 0]
i ← 0
i ≤ ROWS - 1
True
False
j ← 0
j ≤ COLUMNS - 1
True
False
a[i, j] > max
True
False
max ← a[i, j]
j ← j + 1
i ← i + 1
Write
max
End

6. Solution


Start
Const N = 10
i ← 0
i ≤ N - 1
True
False
j ← 0
j ≤ N - 1
True
False
i == j
False
True
i > j
False
True
a[i, j] ← "*"
a[i, j] ← "-"
a[i, j] ← "+"
i ← i + 1
j ← j + 1
i ← 0
i ≤ ROWS - 1
True
False
j ← 0
j ≤ COLUMNS - 1
True
False
Write
a[i, j]
j ← j + 1
i ← i + 1
End

7. Solution



8. Solution

```
for (i = 0; i <= CITIES - 1; i++) {
  do {
    b[i] = Double.Parse(Console.ReadLine());
  } while (b[i] >= 0);
}
```

9. Solution

```
static void Main() {
  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.Write("The End");

  Console.ReadKey();
}
```

10. Solution

```
for (i = 0; i <= PEOPLE - 1; i++) {
  do {
    a[i] = Int32.Parse(Console.ReadLine());
  } while (a[i] % 2 == 2);
}
```

11. 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.Write(a[max_i, max_j]);
```

12. Solution

```
for (i = 0; i <= ELEMENTS - 1; i++) {
  a[i] = Double.Parse(Console.ReadLine());
  while (a[i] < 0) {
    Console.WriteLine("Error");
    a[i] = Double.Parse(Console.ReadLine());
  }
}
```

13. Solution

```
i = 0;
S = 0;
a[i] = Double.Parse(Console.ReadLine());
i++;
while (i < 90) {
  S += a[i - 1] * i;
  a[i] = Double.Parse(Console.ReadLine());
  i++;
}
Console.WriteLine(S);
while (i >= 0) {
  Console.WriteLine(a[i]);
  i -= 5;
}
```

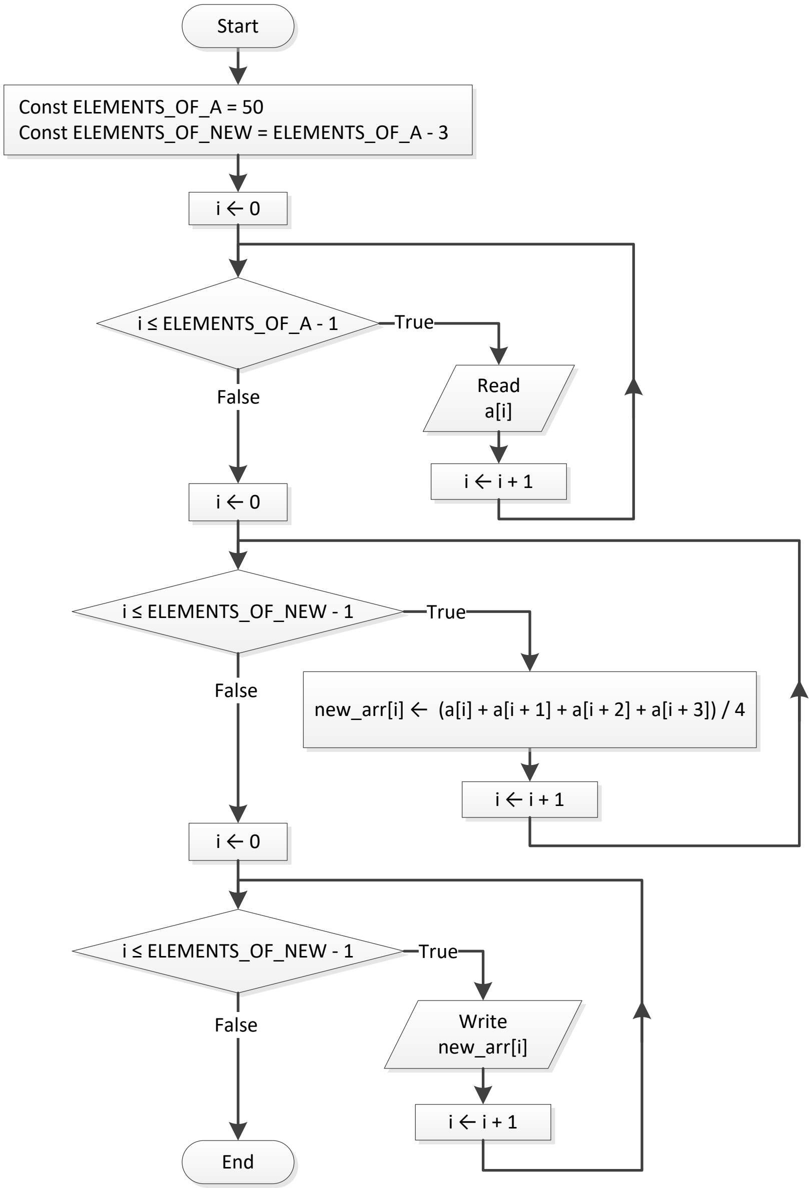
14. Solution

```
for (i = 0; i <= ROWS - 1; i++) {
  max = a[i, 0];
  for (j = 1; j <= COLUMNS - 1; j++) {
    if (a[i, j] > max) {
      max = a[i, j];
    }
  }
  Console.WriteLine(max);
}
```

15. 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());
    }
  }
}
```

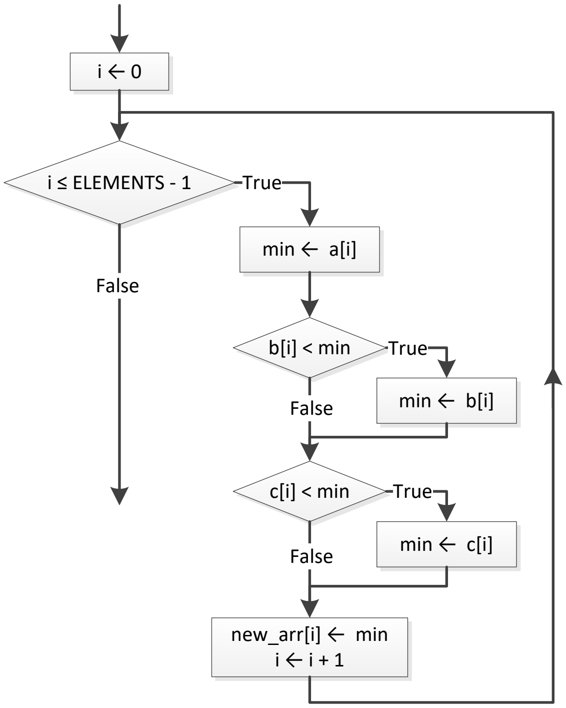
16. Solution



```
const int ELEMENTS_OF_A = 50;
const int ELEMENTS_OF_NEW = ELEMENTS_OF_A - 3;

static void Main() {
  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");
  }
  Console.ReadKey();
}
```

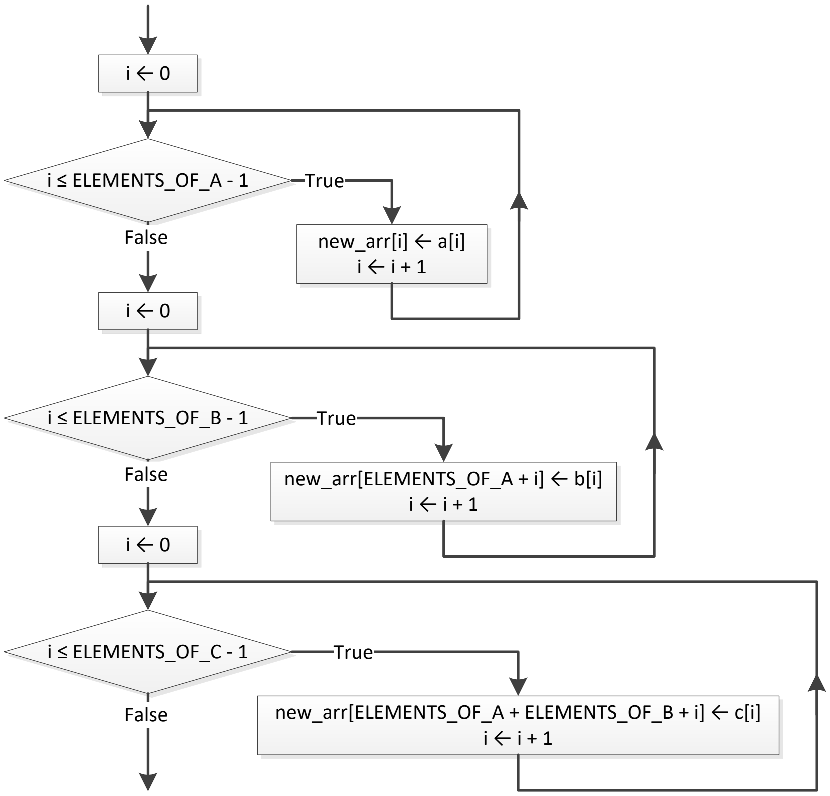
17. Solution

```
const int ELEMENTS = 15;

static void Main() {
  int i;
  double min;

  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++) {
    min = a[i];
    if (b[i] < min) {
      min = b[i];
    }
    if (c[i] < min) {
      min = c[i];
    }
    new_arr[i] = min;
  }

  for (i = 0; i <= ELEMENTS - 1; i++) {
    Console.WriteLine(new_arr[i]);
  }
  Console.ReadKey();
}
```



18. 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() {
  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");
  }
  Console.ReadKey();
}
```

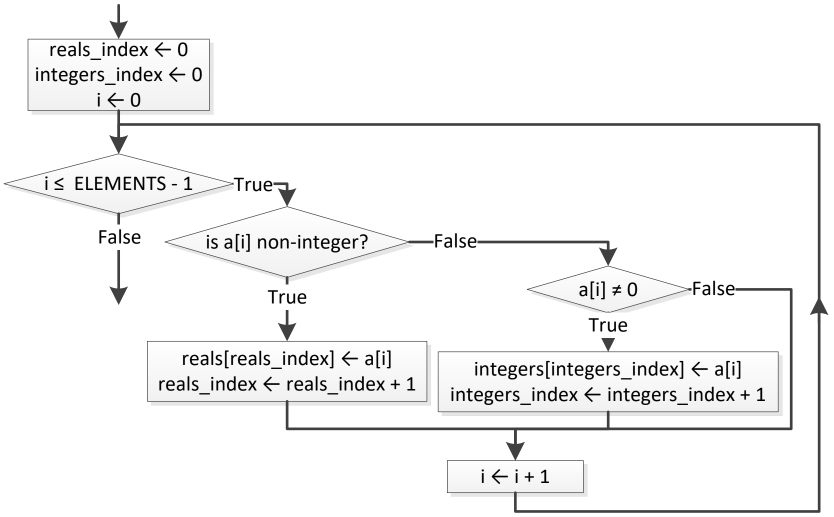


19. Solution

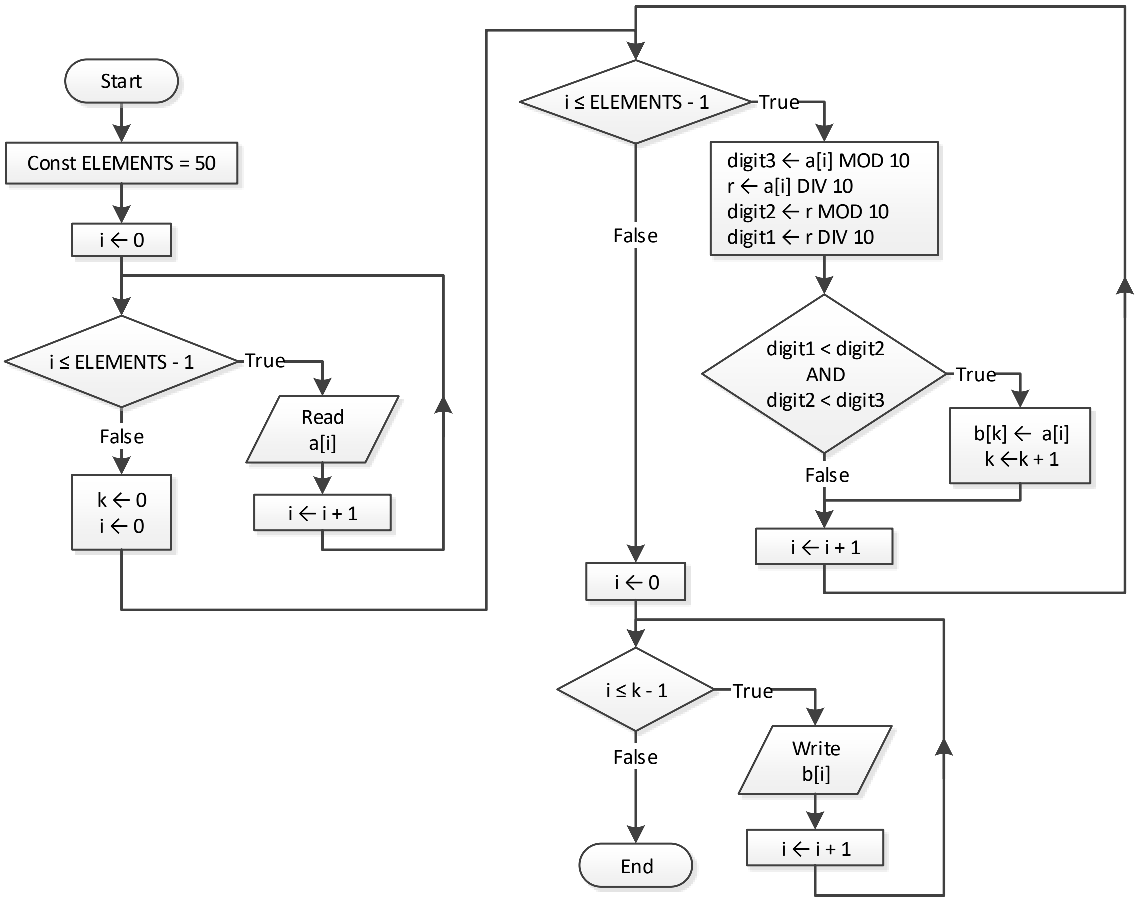
```
const int COLUMNS_OF_A = 10;
const int COLUMNS_OF_B = 15;
const int COLUMNS_OF_C = 20;
const int ROWS = 5;
const int COLUMNS = COLUMNS_OF_A + COLUMNS_OF_B + COLUMNS_OF_C;

static void Main() {
  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();
  }
  Console.ReadKey();
}
```

20. Solution

```
const int ELEMENTS = 50;

static void Main() {
  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");
  }
  Console.ReadKey();
}
```


reals_index ← 0
integers_index ← 0
i ← 0
i ≤ ELEMENTS - 1
True
False
is a[i] non-integer?
False
True
a[i] ≠ 0
False
True
reals[reals_index] ← a[i]
reals_index ← reals_index + 1
integers[integers_index] ← a[i]
integers_index ← integers_index + 1
i ← i + 1

21. Solution



```
const int ELEMENTS = 50;

static void Main() {
  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");
  }
  Console.ReadKey();
}
```

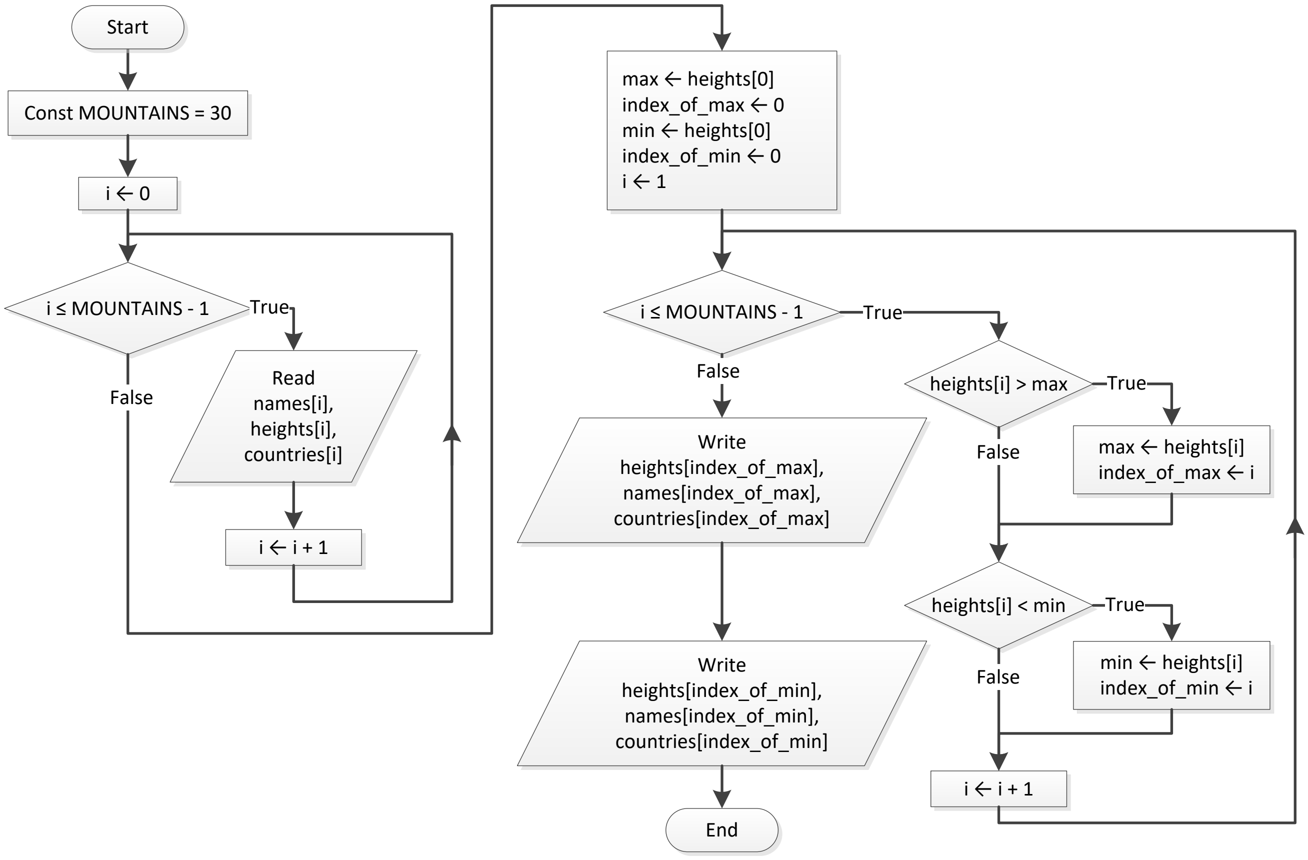
22. Solution

```
const int PRODUCTS = 10;
const int CITIZENS = 1000;

static void Main() {
  int count_B, i, j, max;

  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);
  }

  max = count_A[0];
  for (i = 1; i <= PRODUCTS - 1; i++) {
    if (count_A[i] > max) {
      max = count_A[i];
    }
  }
  for (i = 0; i <= PRODUCTS - 1; i++) {
    if (count_A[i] == max) {
      Console.WriteLine(prod_names[i]);
    }
  }
  Console.ReadKey();
}
```

23. Solution

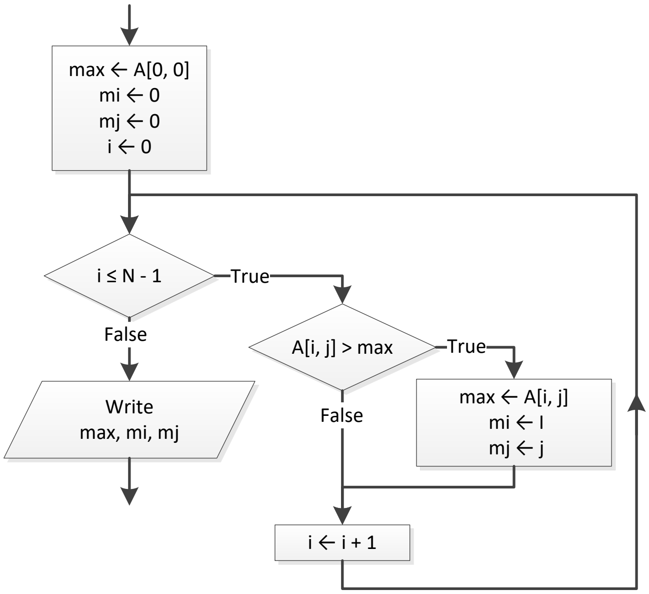
```
const int US_CITIES = 20;
const int CANADIAN_CITIES = 20;

static void Main() {
  int i, j, min_j;
  double min;

  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++) {
    min = distances[i, 0];
    min_j = 0;
    for (j = 1; j <= CANADIAN_CITIES - 1; j++) {
      if (distances[i, j] < min) {
        min = distances[i, j];
        min_j = j;
      }
    }
    Console.WriteLine("Closest Canadian city to " + us_names[i] + " is " + canadian_names[min_j]);
  }
  Console.ReadKey();
}
```

24. Solution



```
const int MOUNTAINS = 30;

static void Main() {
  int i, index_of_max, index_of_min;
  double max, min;

  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();
  }

  max = heights[0];
  index_of_max = 0;
  min = heights[0];
  index_of_min = 0;
  for (i = 1; i <= MOUNTAINS - 1; i++) {
    if (heights[i] > max) {
      max = heights[i];
      index_of_max = i;
    }
    if (heights[i] < min) {
      min = heights[i];
      index_of_min = i;
    }
  }

  Console.Write(heights[index_of_max] + ", " + names[index_of_max] + ", " + countries[index_of_max]);
  Console.WriteLine();
  Console.Write(heights[index_of_min] + ", " + names[index_of_min] + ", " + countries[index_of_min]);
  Console.ReadKey();
}
```

25. Solution



26. Solution

```
const int TEAMS = 26;
const int GAMES = 15;

static void Main() {
  int i, j, m_i, max;

  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;
      }
    }
  }

  max = points[0];
  m_i = 0;
  for (i = 1; i <= TEAMS - 1; i++) {
    if (points[i] > max) {
      max = points[i];
      m_i = i;
    }
  }

  Console.Write(names[m_i]);
  Console.ReadKey();
}
```

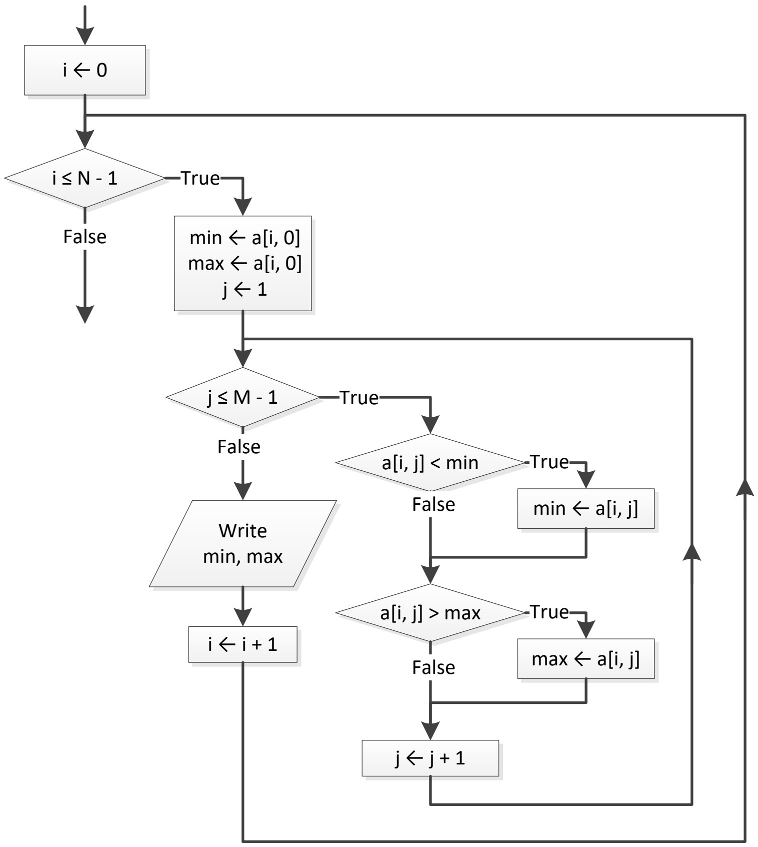
27. Solution

```
const int OBJECTS = 10;
const int FALLS = 20;

static void Main() {
  int i, j;
  double maximum, minimum;

  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[] min = new double[OBJECTS];
  double[] max = new double[OBJECTS];
  for (i = 0; i <= OBJECTS - 1; i++) {
    min[i] = g[i, 0];
    max[i] = g[i, 0];
    for (j = 1; j <= FALLS - 1; j++) {
      if (g[i, j] < min[i]) {
        min[i] = g[i, j];
      }
      if (g[i, j] > max[i]) {
        max[i] = g[i, j];
      }
    }
  }

  for (i = 0; i <= OBJECTS - 1; i++) {
    Console.WriteLine(min[i] + ", " + max[i]);
  }

  maximum = max[0];
  minimum = min[0];
  for (i = 1; i <= OBJECTS - 1; i++) {
    if (max[i] > maximum) {
      maximum = max[i];
    }
    if (min[i] < minimum) {
      minimum = min[i];
    }
  }

  Console.Write(minimum + ", " + maximum);
  Console.ReadKey();
}
```

28. Solution

```
const int STATIONS = 10;
const int DAYS = 365;

static void Main() {
  int i, j, m_i;
  double min;

  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;
  }

  min = average[0];
  m_i = 0;
  for (i = 1; i <= STATIONS - 1; i++) {
    if (average[i] < min) {
      min = average[i];
      m_i = i;
    }
  }
  Console.Write(names[m_i]);
  Console.ReadKey();
}
```

29. Solution



30. Solution

```
const int TEAMS = 20;
const int GAMES = 10;

static void Main() {
  int i, j, m, n, temp;
  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++) {
    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;
      }
    }
  }

  Console.WriteLine("Gold: " + names[0]);
  Console.WriteLine("Silver: " + names[1]);
  Console.Write("Bronze: " + names[2]);
  Console.ReadKey();
}
```

31. Solution

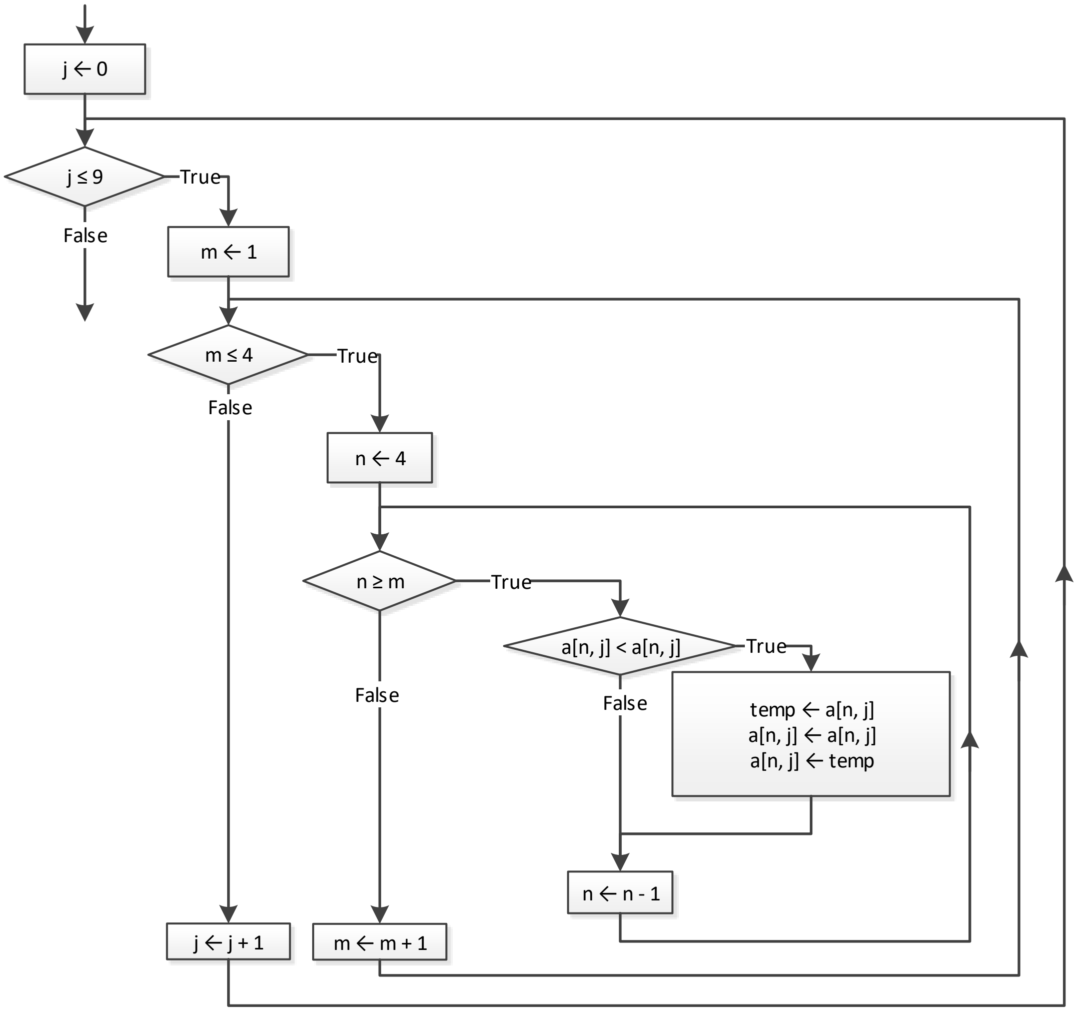
```
const int PEOPLE = 50;
```

```
static void Main() {
  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]);
  }
  Console.ReadKey();
}
```

32. Solution

```
const int ARTISTS = 12;
const int JUDGES = 10;

static void Main() {
  int i, j, m, max, min, 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[] sum = new int[ARTISTS];
  for (i = 0; i <= ARTISTS - 1; i++) {
    sum[i] = 0;
    for (j = 1; j <= JUDGES - 1; j++) {
      sum[i] += score[i, j];
    }
  }

  for (i = 0; i <= ARTISTS - 1; i++) {
    min = score[i, 0];
    max = score[i, 0];
    for (j = 1; j <= JUDGES - 1; j++) {
      if (score[i, j] < min) {
        min = score[i, j];
      }
      if (score[i, j] > max) {
        max = score[i, j];
      }
    }
    sum[i] = sum[i] - min - max;
    Console.WriteLine(sum[i]);
  }

  for (m = 1; m <= ARTISTS - 1; m++) {
    for (n = ARTISTS - 1; n >= m; n--) {
      if (sum[n] > sum[n - 1]) {
        temp = sum[n];
        sum[n] = sum[n - 1];
        sum[n - 1] = temp;

        temp_str = artist_names[n];
        artist_names[n] = artist_names[n - 1];
        artist_names[n - 1] = temp_str;
      }
      else if (sum[n] == sum[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] + ", " + sum[i]);
  }
  Console.ReadKey();
}
```

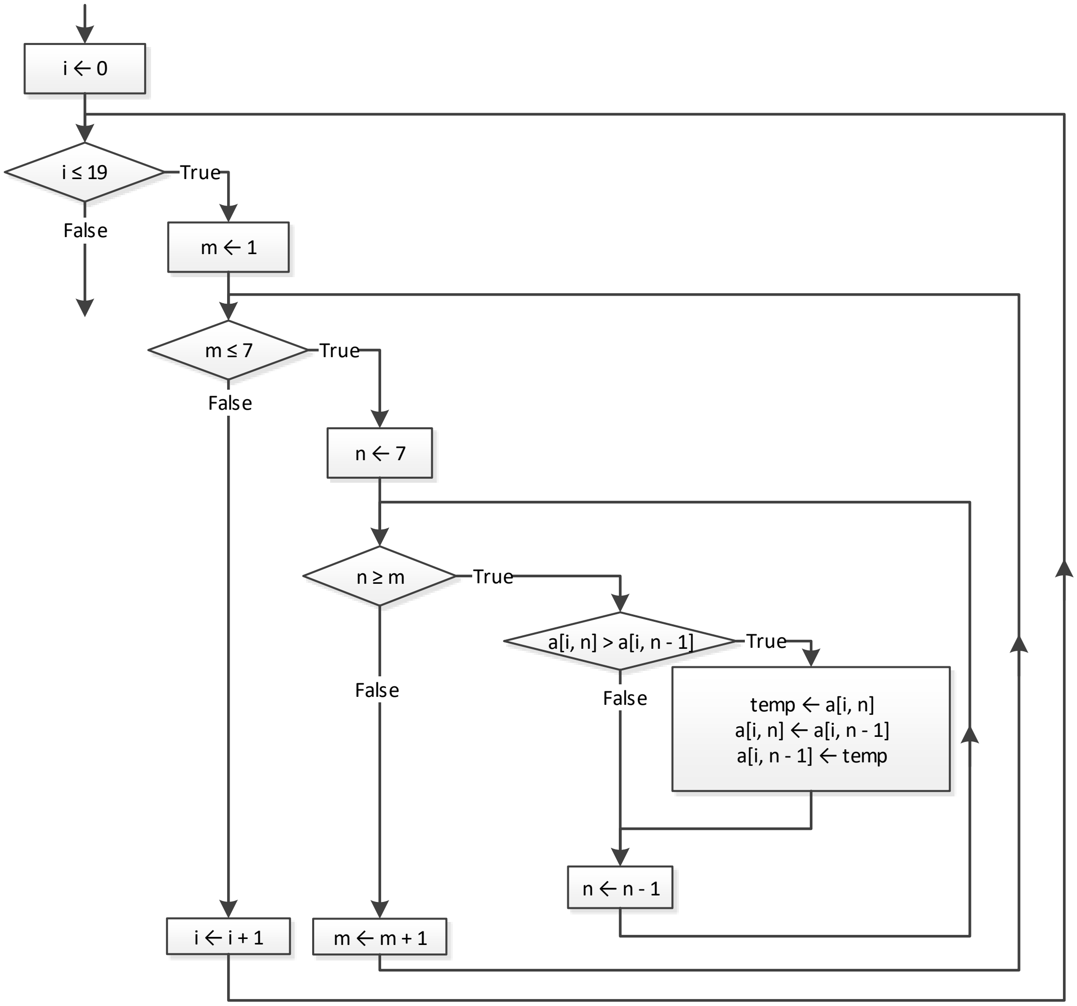
33. Solution



34. Solution

```
const int PEOPLE = 10;
const int PUZZLES = 8;

static void Main() {
  int i, j, m, n;
  double 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 = 1; m <= PUZZLES - 1; m++) {
      for (n = PUZZLES - 1; n >= m; n--) {
        if (times[i, n] < times[i, n - 1]) {
          temp = times[i, n];
          times[i, n] = times[i, n - 1];
          times[i, n - 1] = 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 = 1; m <= PEOPLE - 1; m++) {
    for (n = PEOPLE - 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;
      }
    }
  }

  Console.Write(names[0] + ", " + names[1] + ", " + names[2]);
  Console.ReadKey();
}
```

35. Solution

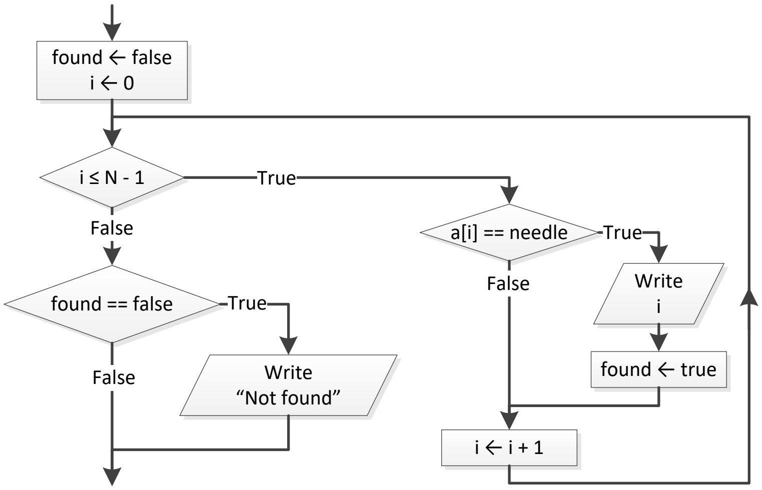


36. Solution

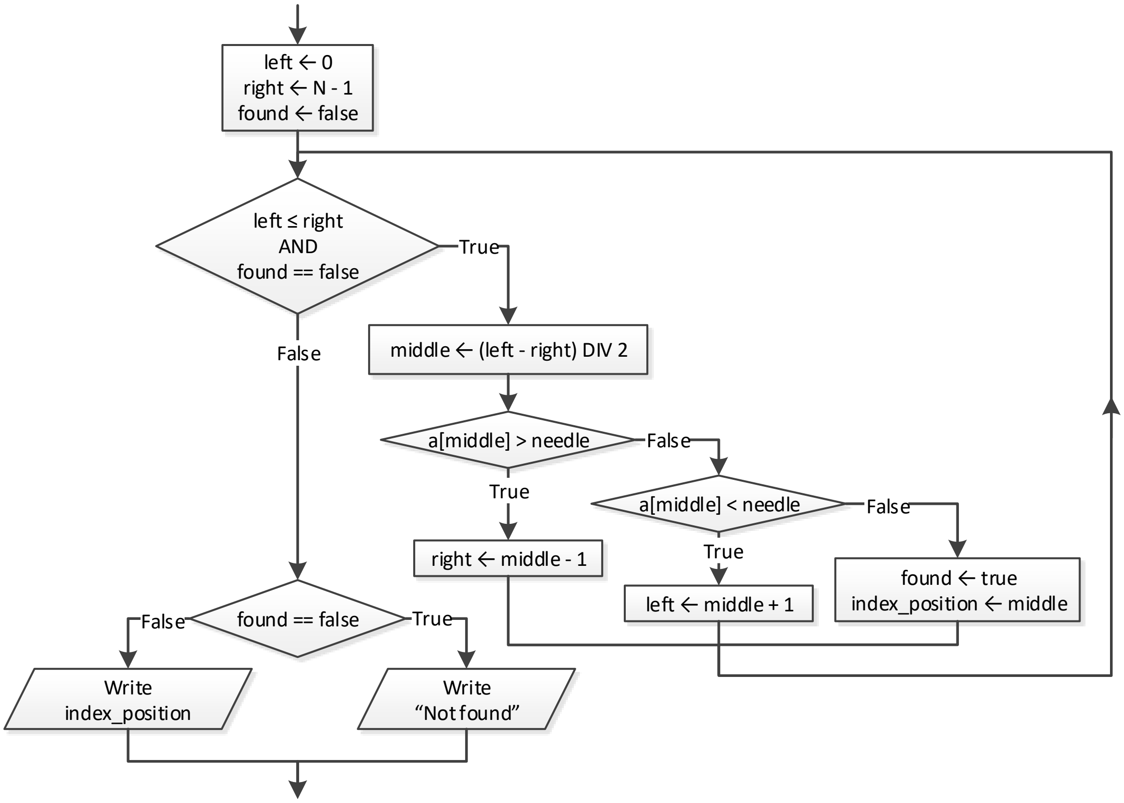
```
const int CITIES = 5;
const int HOURS = 48;

static void Main() {
  int i, j, m, m_i, m_j, n;
  double max, temp;
  string temp_str;

  string[] names = new string[CITIES];
  double[,] CO2 = new double[CITIES, HOURS];
  for (i = 0; i <= CITIES - 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[CITIES];
for (i = 0; i <= CITIES - 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 <= CITIES - 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 <= CITIES - 1; i++) {
    average_per_city[j] += CO2[i, j];
  }
  average_per_city[j] /= CITIES;
}

for (j = 0; j <= HOURS - 1; j++) {
  Console.WriteLine(average_per_city[j]);
}

max = average_per_city[0];
m_j = 0;
for (j = 1; j <= HOURS - 1; j++) {
  if (average_per_city[j] > max) {
    max = average_per_city[j];
    m_j = j;
  }
}
Console.WriteLine(m_j);

max = CO2[0, 0];
m_i = 0;
m_j = 0;
for (i = 0; i <= CITIES - 1; i++) {
  for (j = 0; j <= HOURS - 1; j++) {
    if (CO2[i, j] > max) {
      max = CO2[i, j];
      m_i = i;
      m_j = j;
    }
  }
}
Console.WriteLine(m_j + ", " + names[m_i]);

for (m = 1; m <= CITIES - 1; m++) {
  for (n = CITIES - 1; n >= m; n--) {
    if (average_per_hour[n] > average_per_hour[n - 1]) {
        temp = average_per_hour[n];
        average_per_hour[n] = average_per_hour[n - 1];
        average_per_hour[n - 1] = temp;

        temp_str = names[n];
        names[n] = names[n - 1];
        names[n - 1] = temp_str;
      }
    }
  }

  Console.WriteLine(names[0] + ", " + names[1]  + ", " +  names[2]);
  Console.ReadKey();
}
```

37. Solution



38. Solution



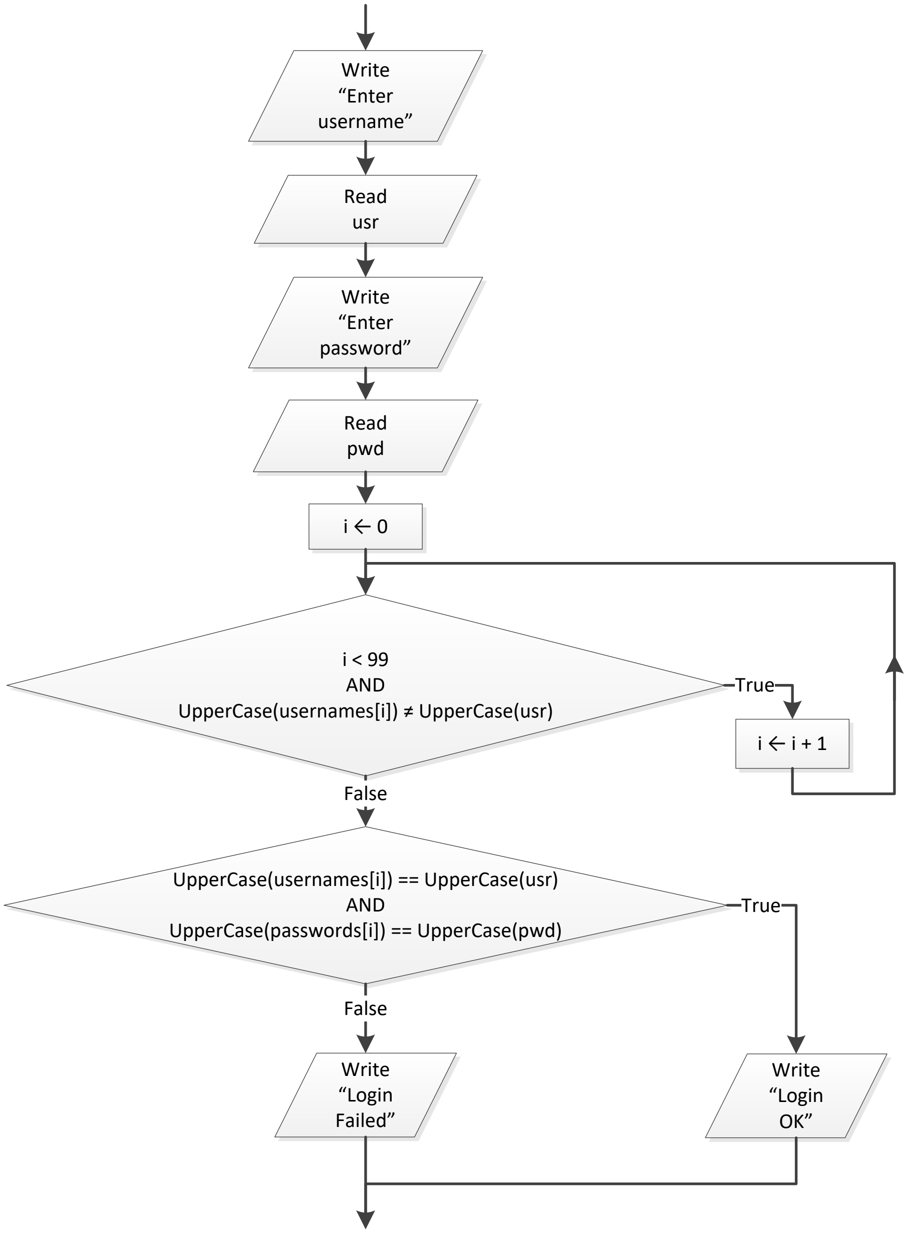
39. Solution

```
const int TEAMS = 10;
const int GAMES = 16;

static void Main() {
  int i, j, sum;
  string needle, input;

  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: ");
      input = Console.ReadLine();
      while (Int32.TryParse(input, out goals_scored[i, j]) == false || goals_scored[i, j] < 0) {
        Console.Write("Error! Enter goals scored: ");
        input = Console.ReadLine();
      }

      Console.Write("Enter goals let in: ");
      input = Console.ReadLine();
      while (Int32.TryParse(input, out goals_let_in[i, j]) == false || goals_let_in[i, j] < 0) {
        Console.Write("Error! Enter goals let in: ");
        input = 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 {
    sum = 0;
    for (j = 0; j <= GAMES - 1; j++) {
      if (goals_scored[i, j] > goals_let_in[i, j]) {
        sum += 3;
      }
      else if (goals_scored[i, j] == goals_let_in[i, j]) {
        sum += 1;
      }
    }
    Console.WriteLine(sum);
  }
  Console.ReadKey();
}
```

40. Solution

```
const int CLASS1 = 20;
const int CLASS2 = 25;

static void Main() {
  int i, left, m, middle, n, right;
  string element, needle;
  bool found;

  Console.WriteLine("Class A");
  string[] names1 = new string[CLASS1];
  for (i = 0; i <= CLASS1 - 1; i++) {
    Console.Write("Enter name: ");
    names1[i] = Console.ReadLine();
  }
  Console.WriteLine("Class B");
  string[] names2 = new string[CLASS2];
  for (i = 0; i <= CLASS2 - 1; i++) {
    Console.Write("Enter name: ");
    names2[i] = Console.ReadLine();
  }

  //Insertion sort algorithm
  for (m = 1; m <= CLASS1 - 1; m++) {
    element = names1[m];
    n = m;
    while (n > 0 && names1[n - 1].CompareTo(element) > 0) {
      names1[n] = names1[n - 1];
      n--;
    }
    names1[n] = element;
  }
  for (m = 1; m <= CLASS2 - 1; m++) {
    element = names2[m];
    n = m;
    while (n > 0 && names2[n - 1].CompareTo(element) > 0) {
      names2[n] = names2[n - 1];
      n--;
    }
    names2[n] = element;
  }

  Console.WriteLine("\nClass A");
  for (i = 0; i <= CLASS1 - 1; i++) {
    Console.WriteLine(names1[i]);
  }
  Console.WriteLine("\nClass B");
  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 == false) {
    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 == true) {
    Console.WriteLine("Student found in class No 1");
  }
  else {
    left = 0;
    right = CLASS2 - 1;
    while (left <= right && found == false) {
      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 == true) {
      Console.WriteLine("Student found in class No 2");
    }
    else {
      Console.WriteLine("Student not found in either class");
    }
  }
  Console.ReadKey();
}
```

41. Solution



```
Console.Write("Enter username: ");
usr = Console.ReadLine();
Console.Write("Enter password: ");
pwd = Console.ReadLine();

i = 0;
while (i < 99 && usernames[i].ToUpper() != usr.ToUpper()) {
  i++;
}
```

```
if (usernames[i].ToUpper() == usr.ToUpper() && passwords[i].ToUpper() == pwd.ToUpper()) {
  Console.WriteLine("Login OK!");
}
else {
  Console.WriteLine("Login Failed!");
}
```

42. Solution

```
Console.Write("Enter a value to search: ");
value_str = Console.ReadLine();
found = false;

//Check if entered value is a valid nine-digit SSN
if (Int32.TryParse(value_str, out value) == true && value >= 100000000 && value <= 999999999) {
  i = 0;
  while (i < 999 && SSNs[i] != value) {
    i++;
  }

  if (SSNs[i] == value) {
    found = true;
    Console.WriteLine(names[i]);
  }
}
else {
  for (i = 0; i <= 999; i++) {
    if (names[i] == value_str) {
      Console.WriteLine(names[i]);
      found = true;
    }
  }
}

if (found == false) {
  Console.WriteLine("This value does not exist");
}
```

43. Solution

```
const int STUDENTS = 12;
const int LESSONS = 6;

static void Main() {
  int i, j;
  bool found;

  int[,] grades = new int[STUDENTS, LESSONS];
  for (i = 0; i <= STUDENTS - 1; i++) {
    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++) {
    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;
    }
  }

  if (found == true) {
    Console.WriteLine("There is at least one student that has an average value below 70");
  }
  Console.ReadKey();
}
```

Chapter 37

37.4 Review Questions: True/False

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

Chapter 38

38.5 Review Questions: True/False

1. true
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. true
18. true
19. false
20. true
21. true
22. true

38.6 Review Exercises

1. Solution

```
static int find_max(int a, int b) {
  int max;
  if (a > b) {
    max = a;
  }
  else {
    max = b;
  }
  return max;
}
```

2. Solution

Step	Statement	Main Code		Function 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.Write(s)`	24 is displayed				

3. Solution

Step	Statement	Main Code		Function sss()		
		s	i	a	sum	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	`sum = 0`			2	**0**	?
11	`k = 1`			2	0	**1**
12	`k <= a`			True		
13	`sum += k`			2	**1**	1
14	`k++`			2	1	**2**
15	`k <= a`			True		
16	`sum += k`			2	**3**	2
17	`k++`			2	3	**3**
18	`k <= a`			False		
19	`return sum`	**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	`sum = 0`			4	**0**	?
29	`k = 1`			4	0	**1**
30	`k <= a`			True		
31	`sum += k`			4	**1**	1
32	`k++`			4	1	**2**
33	`k <= a`			True		
34	`sum += k`			4	**3**	2
35	`k++`			4	3	**3**
36	`k <= a`			True		
37	`sum += k`			4	**6**	4
38	`k++`			4	6	**4**
39	`k <= a`			True		
40	`sum += k`			4	**10**	4
41	`k++`			4	10	**5**
42	`k <= a`			False		
43	`return sum`	**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.Write(s)`	16 is displayed				

4. Solution

Step	Statement	Main Code				Function `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.Write(m + " " + a + " " + x)`	2 1 4 is displayed					
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.Write(m + " " + a + " " + x)`	3 3 9 is displayed					
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.Write(m + " " + a + " " + x)`	4 5 13 is displayed					
23	`a += 2`	12	4	**7**	13		
24	`m++`	12	**5**	7	13		
25	`while (a < 6)`	False					

5. 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;
}
```

6. Solution

```
static double find_min(double a, double b) {
  double min;

  min = a;
  if (b < min) {
    min = b;
  }
  return min;
}

static void Main() {
  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.Write(find_min(find_min(x1, x2), find_min(x3, x4)));
  Console.ReadKey();
}
```

7. 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() {
  double k;
  Console.Write("Enter a temperature in degrees Kelvin: ");
  k = Double.Parse(Console.ReadLine());
  Console.WriteLine("Fahrenheit: " + Kelvin_to_Fahrenheit(k));
  Console.Write("Celsius: " + Kelvin_to_Celsius(k));
  Console.ReadKey();
}
```

8. 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() {
  double height, weight;
  int age;
  string input;

  Console.Write("Enter your weight (in pounds): ");
  input = Console.ReadLine();
  while (Double.TryParse(input, out weight) == false || weight < 0) {
    Console.Write("Error! Enter your weight (in pounds): ");
    input = Console.ReadLine();
  }

  Console.WriteLine("Enter your age: ");
  input = Console.ReadLine();
  while (Int32.TryParse(input, out age) == false || age < 18) {
    Console.Write("Error! Enter your age: ");
    input = Console.ReadLine();
  }

  Console.WriteLine("Enter your height (in inches): ");
  input = Console.ReadLine();
  while (Double.TryParse(input, out height) == false || height < 0) {
    Console.WriteLine("Error! Enter your height (in inches): ");
    input = Console.ReadLine();
  }

  Console.WriteLine(bmi(weight, height));
  Console.ReadKey();
}
```

Chapter 39

39.5 Review Questions: True/False

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

39. 6 Review Exercises

1. Solution

Step	Statement	Main Code		Function `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.Write(a + " is odd")`	The message “3 is odd” is displayed		
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.Write(a + " is odd")`	The message “7 is odd” is displayed		
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.Write(a + " is odd")`	The message “9 is odd” is displayed		
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.Write(a + " is even")`	The message “2 is even” is displayed		
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.Write(a + " is even")`	The message “4 is even” is displayed		
31	`i++`	**6**	4	
32	`i <= 5`	False		

2. *Solution*

Step	Statement	Main Code		Function `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.Write(a * b)`	0 is displayed			
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.Write(a * b)`	93 is displayed			
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.Write(a * b)`	96 is displayed			
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.Write(a * b)`	99 is displayed			
25	`x = 4 * y`	**132**	33		
26	`y++`	132	**34**		
27	`while (x % y < 30)`	False			

3. Solution

Step	Statement	Main Code		Function `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.Write(s)`	20 is displayed				
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.Write(s)`	56 is displayed				
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.Write(s)`	120 is displayed				
52	`i++`	**4**	4			
53	`i <= 3`	False				

4. Solution

```
static void maximum(double a, double b, double c, double d, double e) {
  double max;

  max = a;
  if (b > max) {
    max = b;
  }
  if (c > max) {
    max = c;
  }
  if (d > max) {
    max = d;
  }
  if (e > max) {
    max = e;
  }
  Console.WriteLine(max);
}
```

5. 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() {
  int m, y;

  Console.Write("Enter a year: ");
  y = Int32.Parse(Console.ReadLine());
  for (m = 1; m <= 12; m++) {
    num_of_days(y, m);
  }
  Console.ReadKey();
}
```

6. 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() {
  int choice;
  double distance;

  do {
    display_menu();

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

    if (choice == 3) {
      Console.WriteLine("Bye!");
    }
    else {
      Console.WriteLine("Enter distance: ");
      distance = Double.Parse(Console.ReadLine());
      if (choice == 1) {
        meters_to_miles(distance);
      }
      else {
        miles_to_meters(distance);
      }
    }
  } while (choice != 3);
}
```

7. 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() {
  int seconds;

  Console.Write("Enter number of seconds: ");
  seconds = Int32.Parse(Console.ReadLine());
  amount_to_pay(seconds);
  Console.ReadKey();
}
```

Chapter 40

40.9 Review Questions: True/False

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

40.10 Review Exercises

1. Solution

The value 5 is displayed

2. Solution

The value 14 is displayed

3. Solution

The value 14 is displayed

4. Solution

Step	Statement	Main Code				Function swap()		
		a	m	k	x	x	y	temp
1	k = Int32.Parse(Console.ReadLine())	?	?	**12**	?			
2	m = 1	?	**1**	12	?			
3	a = 1	**1**	1	12	?			
4	while (a < 8)	True						
5	if (k % m != 0)	False						
6	x = a + m + (int)(a - m)	1	1	12	**2**			
7	Console.Write(m + " " + a + " " + x)	1 1 2 is displayed						
8	a += 2	**3**	1	12	2			
9	m++	3	**2**	12	2			

10	`swap(a, m)`					**3**	**2**	?
11	`temp = x`					3	2	**3**
12	`x = y`					**2**	2	3
13	`y = temp`					2	**3**	3
14	`while (a < 8)`	**2**	**3**	12	2			
		True						
15	`if (k % m != 0)`	False						
16	`x = a + m + (int)(a - m)`	2	3	12	**4**			
17	`Console.Write(m + " " + a + " " + x)`	3 2 4 is displayed						
18	`a += 2`	**4**	3	12	4			
19	`m++`	4	**4**	12	4			
20	`swap(a, m)`					**4**	**4**	?
21	`temp = x`					4	4	**4**
22	`x = y`					**4**	4	4
23	`y = temp`					4	**4**	4
24	`while (a < 8)`	**4**	**4**	12	4			
		True						
25	`if (k % m != 0)`	False						
26	`x = a + m + (int)(a - m)`	4	4	12	**8**			
27	`Console.Write(m + " " + a + " " + x)`	4 4 8 is displayed						
28	`a += 2`	**6**	4	12	8			
29	`m++`	6	**5**	12	8			
30	`swap(a, m)`					**6**	**5**	?
31	`temp = x`					6	5	**6**
32	`x = y`					**5**	5	6
33	`y = temp`					5	**6**	5
34	`while (a < 8)`	**5**	**6**	12	8			
		True						
35	`if (k % m != 0)`	False						
36	`x = a + m + (int)(a - m)`	5	6	12	**10**			
37	`Console.Write(m + " " + a + " " + x)`	6 5 10 is displayed						
38	`a += 2`	**7**	6	12	10			
39	`m++`	7	**7**	12	10			
40	`swap(a, m)`					**7**	**7**	?
41	`temp = x`					7	7	**7**

42	`x = y`					**7**	7	7
43	`y = temp`					7	**7**	7
44	`while (a < 8)`	**7**	**7**	12	10			
		True						
45	`if (k % m != 0)`	True						
46	`x = a % m`	7	7	12	**0**			
47	`swap(m, a)`					**7**	**7**	?
48	`temp = x`					7	7	**7**
49	`x = y`					**7**	7	7
50	`y = temp`					7	**7**	7
51	`Console.Write(m + " " + a + " " + x)`	**7**	**7**	12	0			
		7 7 0 is displayed						
52	`a += 2`	**9**	7	12	0			
53	`m++`	9	**8**	12	0			
54	`swap(a, m)`					**9**	**8**	?
55	`temp = x`					9	8	**9**
56	`x = y`					**8**	8	9
57	`y = temp`					8	**9**	9
58	`while (a < 8)`	**8**	**9**	12	0			
		False						

5. Solution

"hellohellohello" is displayed

6. Solution

The value 15 is displayed

7. Solution

11 4 is displayed

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.Write("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() {
  int i;

  string[] names = new string[STUDENTS];
  int[,] grades = new int[STUDENTS, LESSONS];
  part1(names, grades);

  double[] average = new double[STUDENTS];
  average = part2(grades);

  part3(average, names);

  for (i = 0; i <= STUDENTS - 1; i++) {
    Console.WriteLine(names[i] + "\t" + average[i]);
  }
  Console.ReadKey();
}
```

9. Solution

```
static string part1() {
  string message;
  Console.Write("Enter a message: ");
  message = Console.ReadLine().ToLower();
  return message;
}

static int part2(string message) {
  int last_pos = message.Length - 1;
  return last_pos;
}

static string part3(string message, int last_pos) {
  string letter, message_clean;
  int i;

  message_clean = "";
  for (i = 0; i <= last_pos; i++) {
    letter = message[i].ToString();
    if (letter != " " && letter != "," && letter != "." && letter != "?") {
      message_clean += letter;
    }
  }
  return message_clean;
}

static bool part4(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; //or you can write  j = part2(message_clean);
  palindrome = true;
  for (i = 0; i <= middle_pos; i++) {
    left_letter = message_clean[i].ToString();
    right_letter = message_clean[j].ToString();
    if (left_letter != right_letter) {
      palindrome = false;
      break;
    }
    j--;
  }
  return palindrome;
}

static bool part5(string message) {
  int last_pos;
  string message_clean;
  bool palindrome;

  last_pos = part2(message);
  message_clean = part3(message, last_pos);
  palindrome = part4(message_clean);
  return palindrome;
}

static void Main() {
  string message;
  bool palindrome;

  message = part1();
  palindrome = part5(message);
  if (palindrome == true) {
    Console.WriteLine("The message is palindrome");
  }
  Console.ReadKey();
}
```

10. Solution

```
static void Main() {
  int a, b, c, max;

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

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

  Console.Write(max);
  Console.ReadKey();
}
```

11. Solution

```
static void f1(double a, double b, double c, ref double sum, ref double average) {
  sum = a + b + c;
  average = sum / 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 string get_input() {
  string answer;

  do {
    Console.Write("Enter Yes or No: ");
    answer = Console.ReadLine().ToUpper();
  } while (answer != "YES" && answer != "NO");
  return answer;
}

static double find_area(double b, double h) {
  return b * h;
}

static void Main() {
  double bas, height;

  do {
    Console.Write("Enter the base of the parallelogram: ");
    bas = Double.Parse(Console.ReadLine());
    Console.Write("Enter the height of the parallelogram: ");
    height = Double.Parse(Console.ReadLine());

    Console.WriteLine("Area = " + find_area(bas, height));

    Console.WriteLine("Would you like to repeat? ");
  } while (get_input() == "YES");
}
```

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, sum = 0;

  for (i = 0; i <= STUDENTS - 1; i++) {
    sum += grades[i];
  }
  return sum / (double)STUDENTS;
}

static void sort_arrays(int[] grades, string[] names) {
  int m, n, temp;
  string temp_str;

  for (m = 1; m <= STUDENTS - 1; m++) {
    for (n = STUDENTS - 1; n >= m; n--) {
        if (grades[n] > grades[n - 1]) {
          temp = grades[n];
          grades[n] = grades[n - 1];
          grades[n - 1] = temp;

          temp_str = names[n];
          names[n] = names[n - 1];
          names[n - 1] = temp_str;
        }
    }
  }
}

static void Main() {
  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]);
    }
  }
  Console.ReadKey();
}
```

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 min, ref int max) {
  int i;

  min = score[0];
  max = score[0];
  for (i = 1; i <= JUDGES - 1; i++) {
    if (score[i] > max) {
      max = score[i];
    }
    if (score[i] < min) {
      min = score[i];
    }
  }
}

static void Main() {
  string name;
  int sum, i, points, min = 0, max = 0;
  int[] score = new int[JUDGES];

  Console.Write("Enter artist's name: ");
  name = Console.ReadLine();
  score = get_array();
  find_min_max(score, ref min, ref max);

  sum = 0;
  for (i = 0; i <= JUDGES - 1; i++) {
    sum += score[i];
  }

  points = sum - min - max;
  Console.Write("Artist " + name + " got " + points + " points");
  Console.ReadKey();
}
```

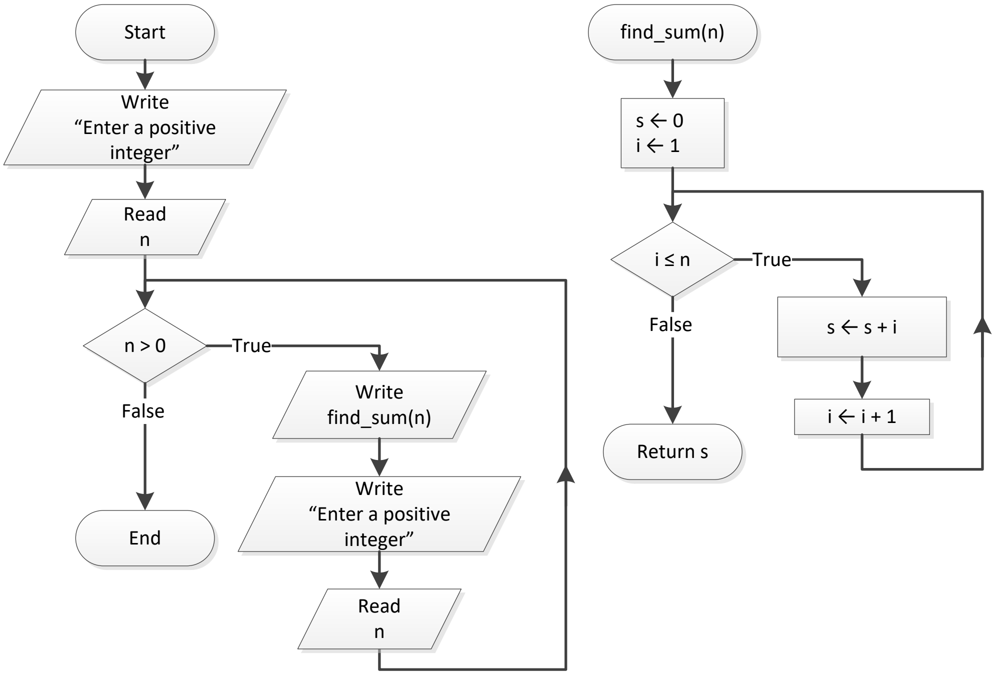
16. Solution

```
static double woc(int index) {
  int i;
  double number_of_grains;

  number_of_grains = 1;
  for (i = 1; i <= index - 1; i++) {
    number_of_grains *= 2;
  }
  return number_of_grains;
}

static void Main() {
  double sum;
  int i;

  sum = 0;
  for (i = 1; i <= 64; i++) {
    sum += woc(i);
  }
  Console.Write(sum);
  Console.ReadKey();
}
```

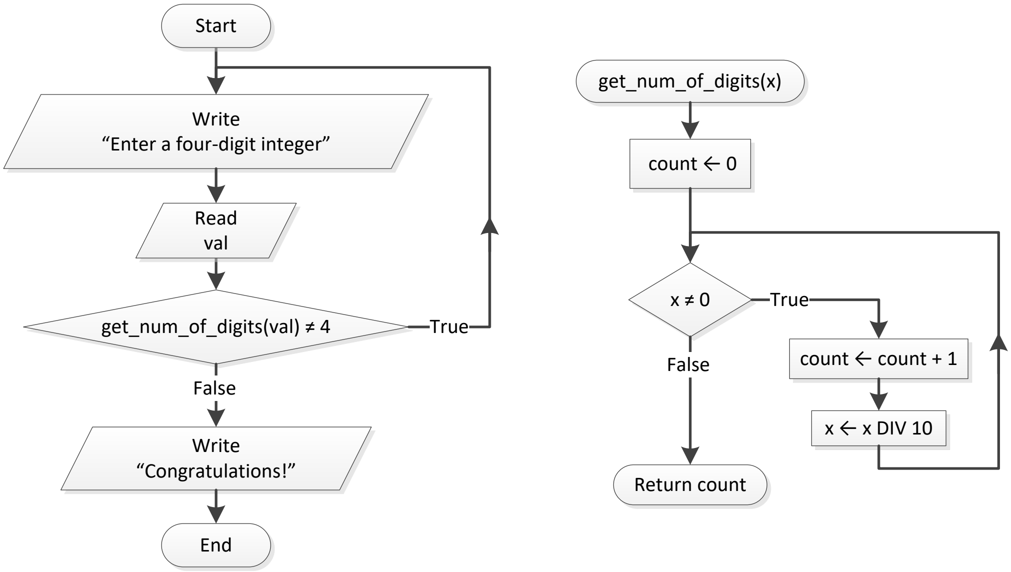
Chapter 41

41.3 Review Exercises

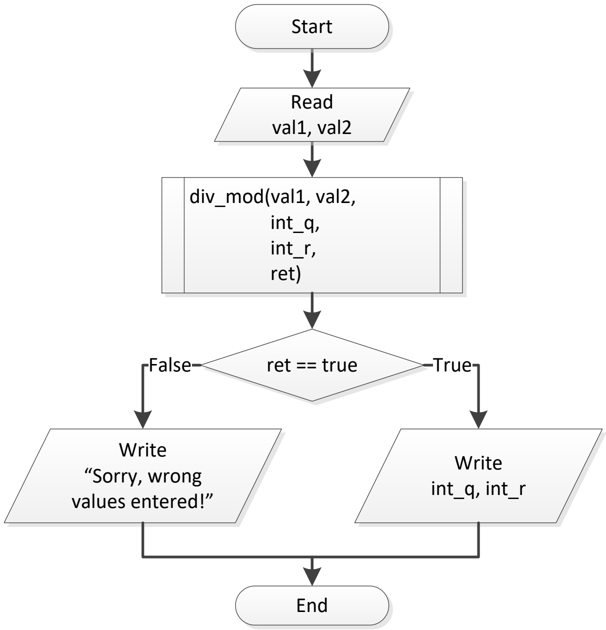
1. Solution

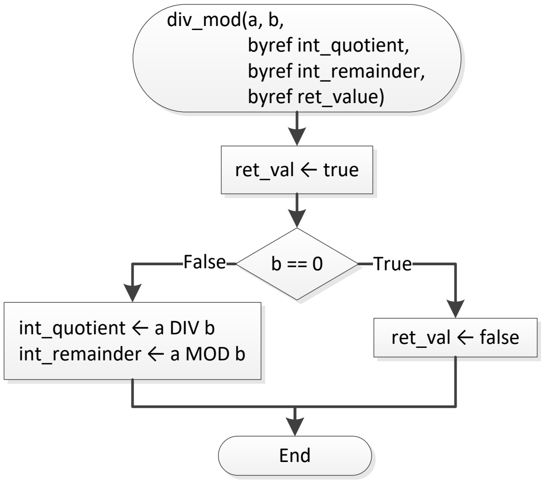


2. Solution



3. Solution

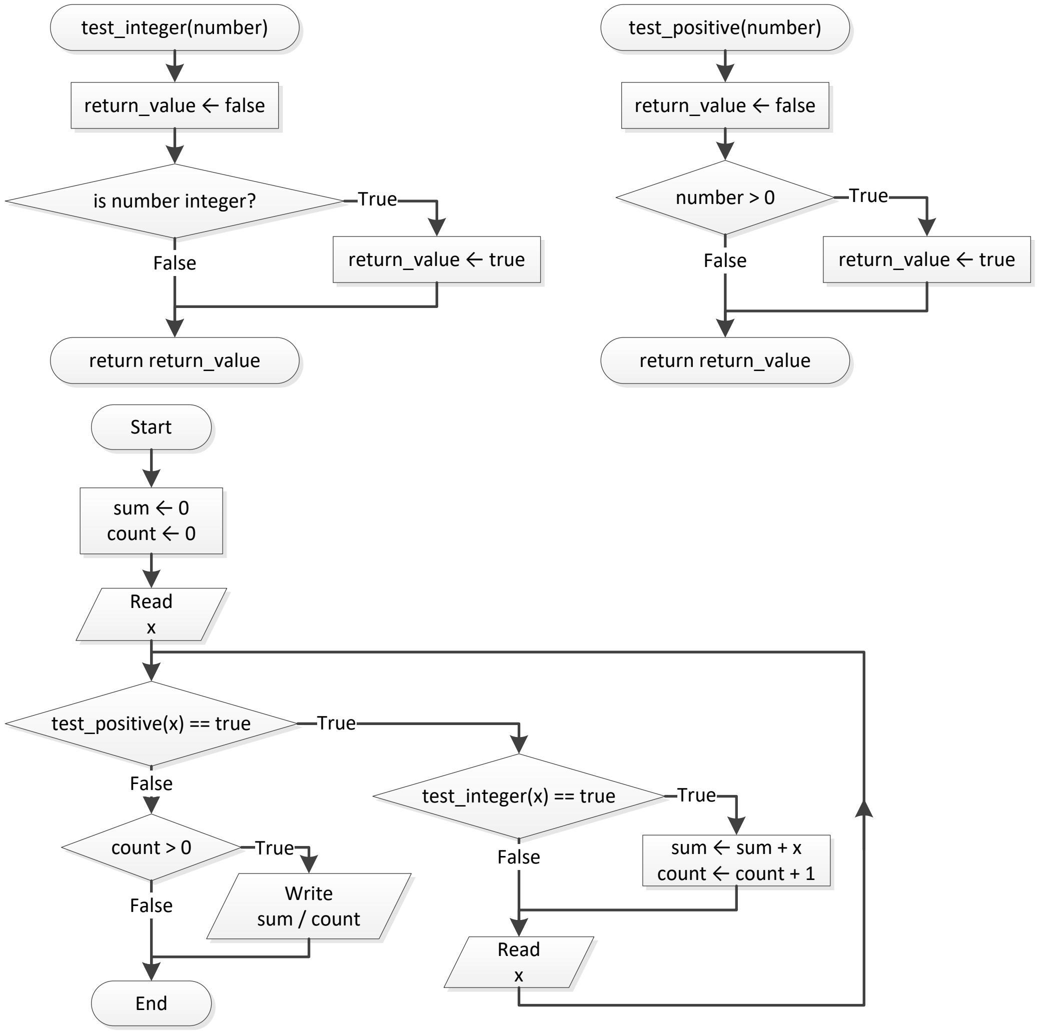




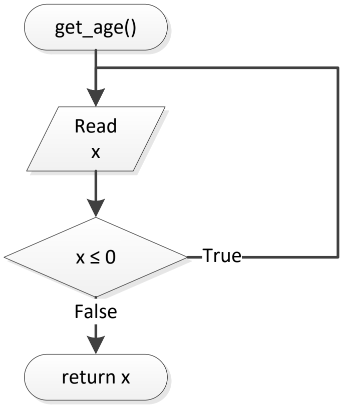
Notice: *Some programmers, instead of using the keyword* `byref`*, prefer to write the keyword* `inout`*, which denotes pretty much the same thing— that the variable is both input (it accepts values) and output (it returns values).*

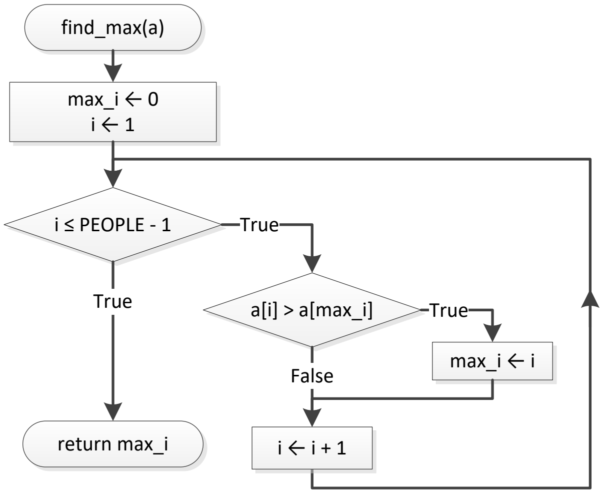
Notice: *Please note that in flowcharts the keyword* `byref` *should be written only within the formal argument list.*

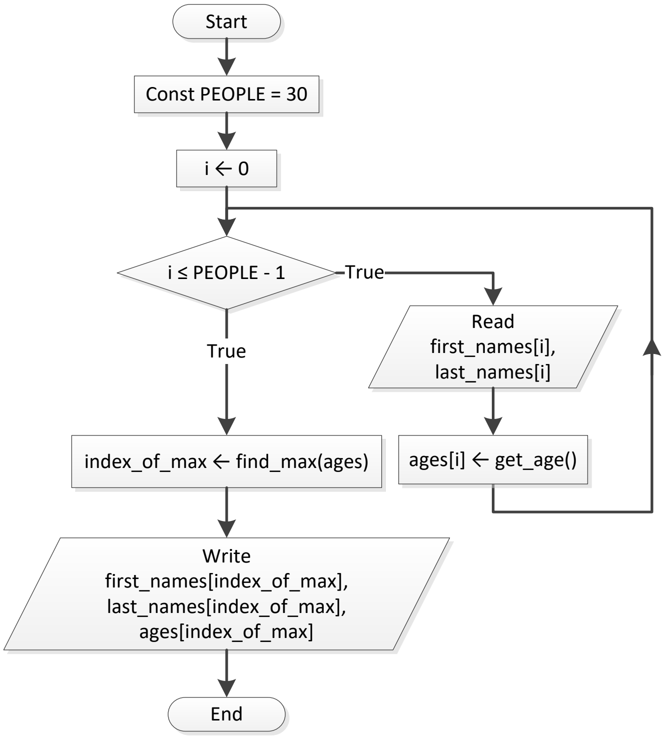
4. Solution


test_integer(number)
return_value ← false
is number integer?
True
return_value ← true
False
return return_value
test_positive(number)
return_value ← false
number > 0
True
return_value ← true
False
return return_value
Start
sum ← 0
count ← 0
Read
x
test_positive(x) == true
True
False
test_integer(x) == true
True
sum ← sum + x
count ← count + 1
False
Read
x
count > 0
True
Write
sum / count
False
End

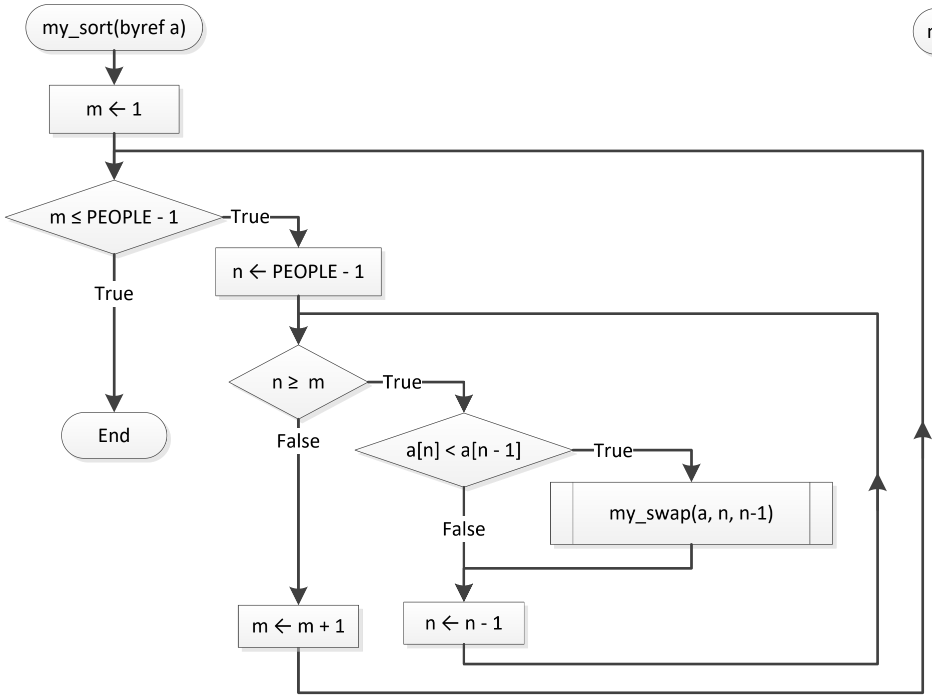
5. Solution

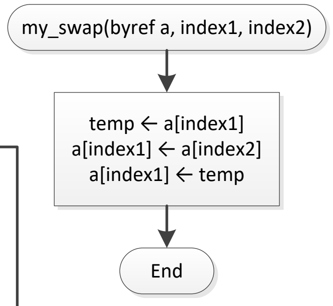

get_age()
Read
x
x ≤ 0
True
False
return x

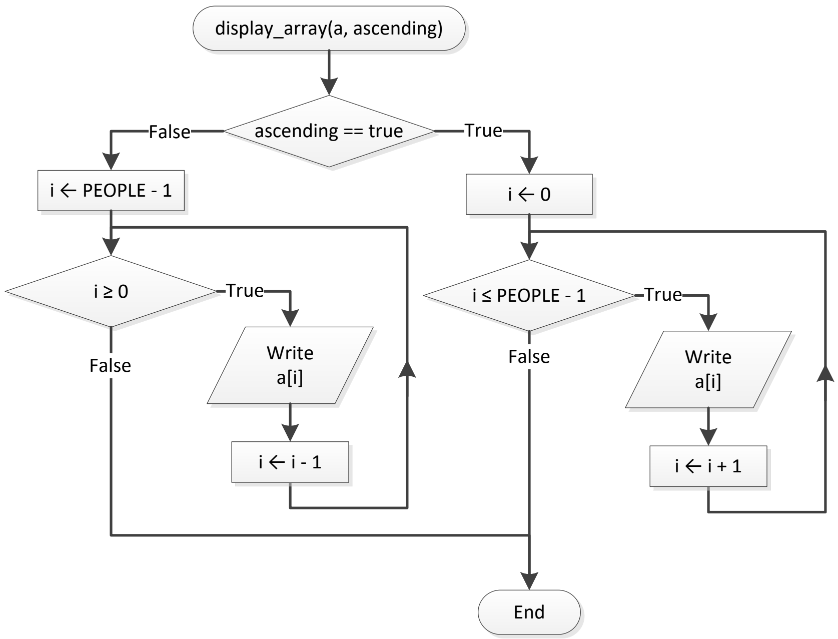

find_max(a)
max_i ← 0
i ← 1
i ≤ PEOPLE - 1
True
True
a[i] > a[max_i]
True
max_i ← i
False
i ← i + 1
return max_i

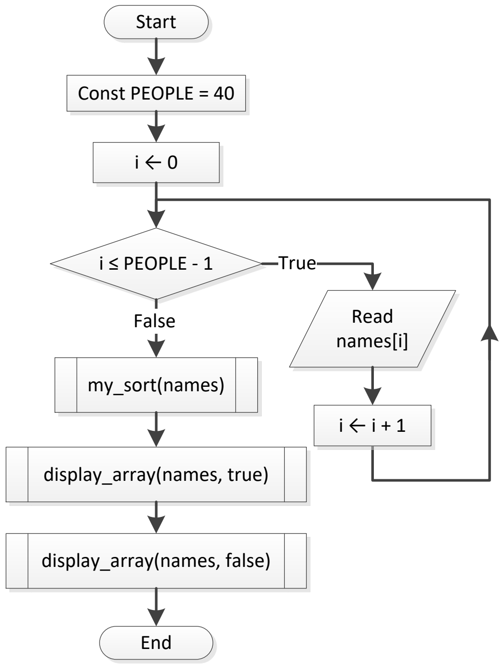

Start
Const PEOPLE = 30
i ← 0
i ≤ PEOPLE - 1
True
Read
first_names[i],
last_names[i]
ages[i] ← get_age()
True
index_of_max ← find_max(ages)
Write
first_names[index_of_max],
last_names[index_of_max],
ages[index_of_max]
End

6. Solution

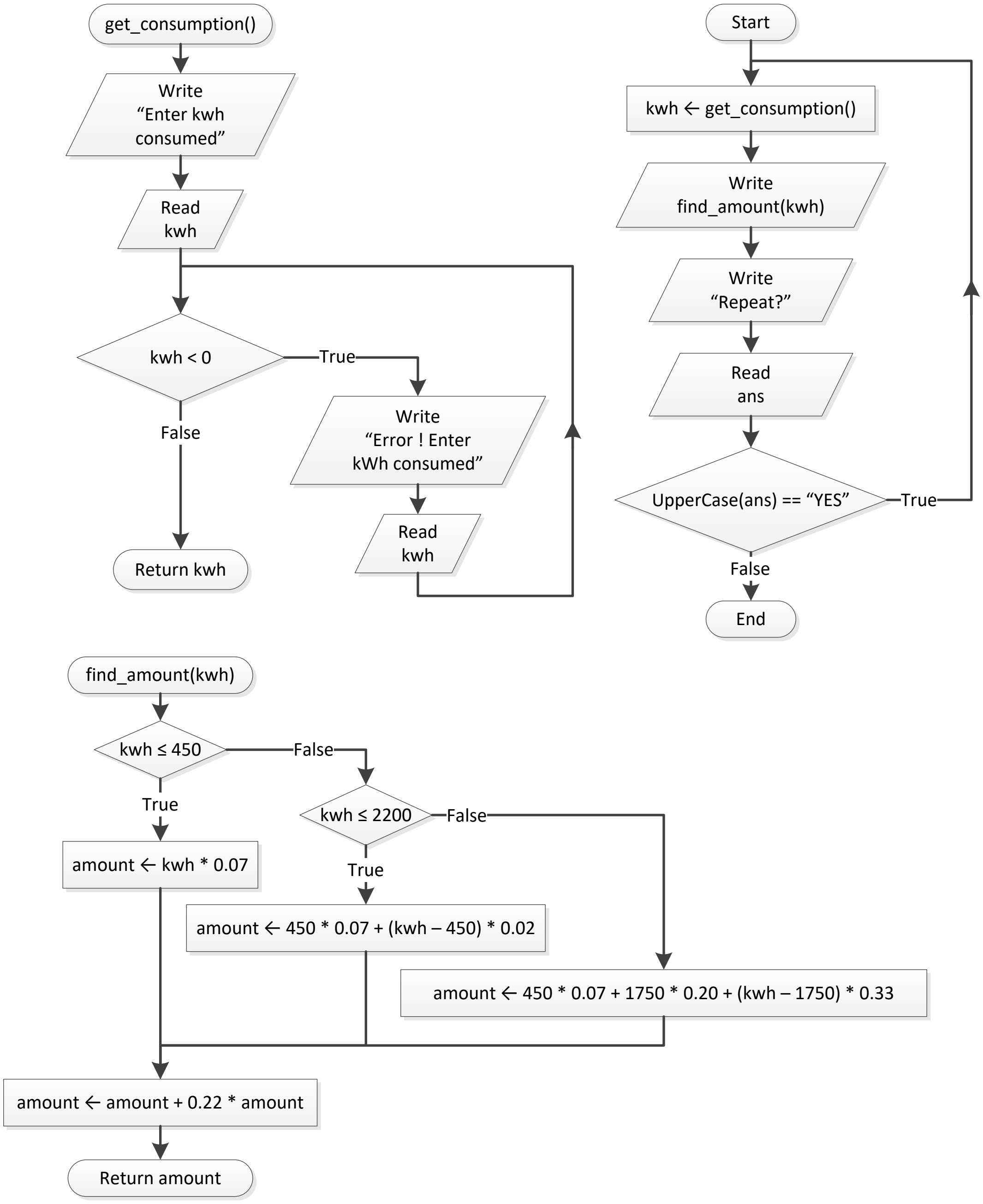

my_sort(byref a)
m ← 1
m ≤ PEOPLE - 1
True
True
End
n ← PEOPLE - 1
n ≥ m
True
False
a[n] < a[n - 1]
True
False
my_swap(a, n, n-1)
m ← m + 1
n ← n - 1


my_swap(byref a, index1, index2)
temp ← a[index1]
a[index1] ← a[index2]
a[index1] ← temp
End


display_array(a, ascending)
ascending == true
False
True
i ← PEOPLE - 1
i ≥ 0
True
Write
a[i]
i ← i - 1
False
i ← 0
i ≤ PEOPLE - 1
True
Write
a[i]
i ← i + 1
False
End


Start
Const PEOPLE = 40
i ← 0
i ≤ PEOPLE - 1
True
Read
names[i]
i ← i + 1
False
my_sort(names)
display_array(names, true)
display_array(names, false)
End

7. Solution


get_consumption()
Write
“Enter kwh
consumed”
Read
kwh
kwh < 0
True
Write
“Error ! Enter
kWh consumed”
Read
kwh
False
Return kwh
Start
kwh ← get_consumption()
Write
find_amount(kwh)
Write
“Repeat?”
Read
ans
UpperCase(ans) == “YES”
True
False
End
find_amount(kwh)
kwh ≤ 450
False
True
amount ← kwh * 0.07
kwh ≤ 2200
False
True
amount ← 450 * 0.07 + (kwh – 450) * 0.02
amount ← 450 * 0.07 + 1750 * 0.20 + (kwh – 1750) * 0.33
amount ← amount + 0.22 * amount
Return amount

8. 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() {
  int val;

  do {
    Console.Write("Enter a positive integer ");
    val = Int32.Parse(Console.ReadLine());
  } while (val < 0);

  Console.Write(f1(val));
  Console.ReadKey();
}
```

Notice: Please note that in the division n / 2.0, the constant value 2.0 is considered a real. In C#, you will get a completely different result if you write n / 2 instead of n / 2.0 when variable n is an integer. The result of the division n / 2 is an integer, while the result of the division n / 2.0 is a real.

9. 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] = Double.Parse(Console.ReadLine());
  }
  return values;
}

static void find_min_max(double[] values, ref int min_i, ref int max_i) {
  int i;

  min_i = 0;
  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() {
  int min_i = 0, max_i = 0;

  double[] v = read_values();
  find_min_max(v, ref min_i, ref max_i);
  Console.Write(v[min_i] + ", " + v[max_i]);
  Console.ReadKey();
}
```

Notice*: Please study the way the procedure* `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 Chapter 36; however, it can be used as an alternative.*

10. Solution

```
const int STUDENTS = 20;
const int LESSONS = 10;

static void get_arrays(string[] names, int[,] grades)  {
  int i, j;

  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());
    }
  }
}

static double[] find_average(int[,] grades) {
  int i, j;
  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;
  }
  return average;
}

static void display(string[] names, double[] average) {
  int i;

  for (i = 0; i <= STUDENTS - 1; i++) {
    if (average[i] > 89) {
      Console.WriteLine(names[i] + ", " + average[i]);
    }
  }
}

static void Main() {
  string[] names = new string[STUDENTS];
  int[,] grades = new int[STUDENTS, LESSONS];
  double[] av = new double[STUDENTS];

  get_arrays(names, grades);
  av = find_average(grades);
  display(names, av);
  Console.ReadKey();
}
```

11. 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) == true) {
        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() {
  int day, month, year, sum, 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) == false) {
    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());
  }

  sum = 0;
  for (i = 1; i <= month - 1; i++) {
    sum += num_of_days(year, i);
  }
  sum += day;

  Console.Write(sum);
  Console.ReadKey();
}
```

12. Solution

```
static Random rnd = new Random();

static int dice() {
  return rnd.Next(1, 7);
}

static void Main() {
  int dice1, dice2, i, player, sum, sum_player1 = 0, sum_player2 = 0;
  string key, name1, name2;
  Console.Write("Player1 - Enter name: ");
  name1 = Console.ReadLine();
  Console.Write("Player2 - Enter name: ");
  name2 = Console.ReadLine();

  for (player = 1; player <= 2; player++) {
    sum = 0;
    for (i = 1; i <= 10; i++) {
      if (player == 1) {
        Console.WriteLine(name1 + ", hit enter to roll the dice!");
      }
      else {
        Console.WriteLine(name2 + ", hit enter to roll the dice!");
      }
      key = Console.ReadLine();

      dice1 = dice();
      dice2 = dice();
      Console.WriteLine(dice1 + "   " + dice2);
      sum += dice1 + dice2;
    }
    if (player == 1) {
      sum_player1 = sum;
    }
    else {
      sum_player2 = sum;
    }
  }

  if (sum_player1 == sum_player2) {
    Console.WriteLine("Tie!");
  }
  else if (sum_player1 > sum_player2) {
    Console.WriteLine(name1 + " wins");
  }
  else {
    Console.WriteLine(name2 + " wins");
  }
  Console.ReadKey();
}
```

13. 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()  {
  int choice;
  Console.WriteLine("1. Gas");
  Console.WriteLine("2. Diesel");
  Console.WriteLine("3. Hybrid");
  Console.Write("Enter type of the car: ");
  choice = Int32.Parse(Console.ReadLine());
  return choice;
}

static int get_days()  {
  int days;
  Console.Write("Enter total number of rental days: ");
  days = Int32.Parse(Console.ReadLine());
  return days;
}

static double get_charge(int type, int rental_days) {
  double charge;

  if (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 (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() {
  int count, i;
  double charge, sum;

  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();
  }

  sum = 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);
     sum += charge;
  }

  count = 0;
  for (i = 0; i <= CARS - 1; i++) {
   if (rented_car_types[i] == HYBRID) {
     count++;
   }
  }

  Console.WriteLine("Hybrids rented: " + count);
  Console.Write("Net profit: " + sum / (1 + TAX_RATE));
  Console.ReadKey();
}
```

14. Solution

```
const int CHANNELS = 10;
const int DAYS = 7;

static void get_data(string[] names, int[,] viewers)  {
  int i, j;

  string[] day_names = {"Monday", "Tuesday", "Wednesday",
                        "Thursday", "Friday", "Saturday", "Sunday"};

  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 sum ,i;

  sum = 0;
  for (i = 0; i <= 4; i++) {
    sum += a[i];
  }
  return sum / 5.0;
}

static void Main() {
  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 == true) {
      Console.WriteLine(names[i]);
    }
  }
  Console.ReadKey();
}
```

15. Solution

```
const int CITIZENS = 30;

static void input_data(int[] SSNs, string[] answers)  {
  int i;

  for (i = 0; i <= CITIZENS - 1; i++) {
    Console.Write("Enter SSN: ");
    SSNs[i] = Int32.Parse(Console.ReadLine());
    Console.Write("Enter answer: ");
    answers[i] = Console.ReadLine();
  }
}

static void sort_arrays(int[] SSNs, string[] answers) {
  int m, n, index_of_min;
  int min, temp;
  string temp_str;

  for (m = 0; m <= CITIZENS - 1; m++) {
    min = SSNs[m];
    index_of_min = m;
    for (n = m; n <= CITIZENS - 1; n++) {
      if (SSNs[n] < min) {
        min = 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(int[] SSNs, int SSN) {
  int left, right, middle, index_position = 0, return_value;
  bool found;

  left = 0;
  right = CITIZENS - 1;
  found = false;
  while (left <= right && found == false) {
    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 == false) {
    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() {
  int[] SSNs = new int[CITIZENS];
  int 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 = Int32.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");
}
```

16. 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 == false) {
      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[] names = new string[TEAMS];
  string[,] results = new string[TEAMS, GAMES];
  int j, index, sum;

  input_data(names, results);
  display_result(names, results);

  index = find_team(names);
  while (index != -1) {
    sum = 0;
    for (j = 0; j <= GAMES - 1; j++) {
      if (results[index, j] == "W") {
        sum += 3;
      }
      else if (results[index, j] == "T") {
        sum += 1;
      }
    }
    Console.WriteLine("Points: " + sum);
    index = find_team(names);
  }
  Console.ReadKey();
}
```

17. Solution

```
//space is a valid character!
static string alphabet = " abcdefghijklmnopqrstuvwxyz";

static string my_encrypt(string message, int encryption_key) {
  string return_value;
  int i, index, new_index;
  string letter, new_letter;

  return_value = "";
  for (i = 0; i <= message.Length - 1; i++) {
    letter = message[i].ToString();
    index = alphabet.IndexOf(letter);
    new_index = (index + encryption_key) % 27; //26 letters + 1 space
    new_letter = alphabet[new_index].ToString();
    return_value += new_letter;
  }
  return return_value;
}

static string my_decrypt(string message, int decryption_key) {
  string return_value;
  int i, index, new_index;
  string letter, new_letter;

  return_value = "";
  for (i = 0; i <= message.Length - 1; i++) {
    letter = message[i].ToString();
    index = alphabet.IndexOf(letter);
    new_index = (index + 27 - decryption_key) % 27; //26 letters + 1 space
    new_letter = alphabet[new_index].ToString();
    return_value += new_letter;
  }
  return return_value;
}

static void display_menu() {
  Console.WriteLine();
  Console.WriteLine("1. Encrypt a message");
  Console.WriteLine("2. Decrypt a message");
  Console.WriteLine("3. Exit");
}

static void Main() {
  int choice, encryption_key, decryption_key;
  string message;

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

    if (choice == 1) {
      Console.Write("Enter a message to encrypt: ");
      message = Console.ReadLine();
      Console.Write("Enter an encryption key: ");
      encryption_key = Int32.Parse(Console.ReadLine());
      Console.WriteLine("Your encrypted message is: " + my_encrypt(message, encryption_key));
    }
    else if (choice == 2) {
      Console.Write("Enter a message to decrypt: ");
      message = Console.ReadLine();
      Console.Write("Enter an decryption key: ");
      decryption_key = Int32.Parse(Console.ReadLine());
      Console.WriteLine("Your decrypted message is: " + my_decrypt(message, decryption_key));
    }
  } while (choice != 3);
}
```

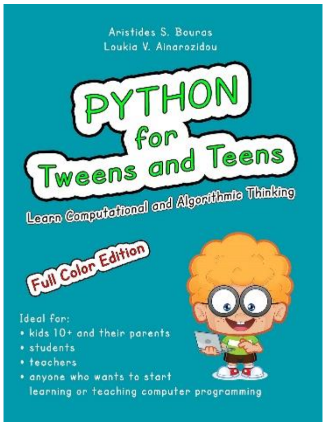
Some Final Words from the Authors

We hope you really enjoyed reading this book. We 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 us your gratitude by writing a good review and giving us as many stars as possible. By doing this, you will encourage us to continue writing and of course you'll help other readers to reach us.

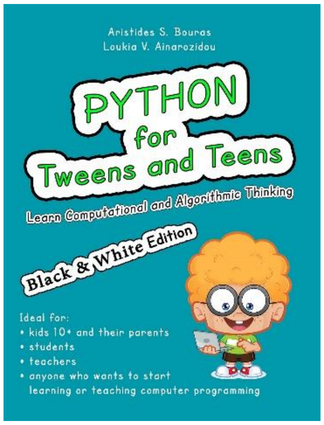
And remember: Learning is a process within an endless loop structure. It begins at birth and continues throughout your lifetime!

More of our Books


Aristides S. Bouras
Loukia V. Ainarozidou
PYTHON for Tweens and Teens
Learn Computational and Algorithmic Thinking
Full Color Edition
Ideal for:
• kids 10+ and their parents
• students
• teachers
• anyone who wants to start learning or teaching computer programming

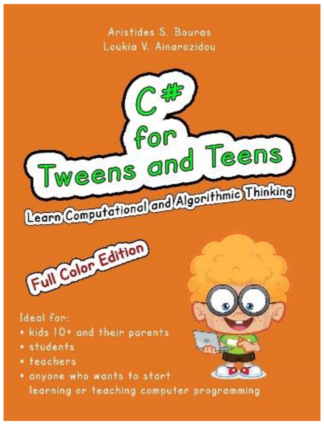
ISBN-10: 1543127940

ISBN-13: 978-1543127942


Aristides S. Bouras
Loukia V. Ainarozidou
PYTHON for Tweens and Teens
Learn Computational and Algorithmic Thinking
Black & White Edition
Ideal for:
• kids 10+ and their parents
• students
• teachers
• anyone who wants to start learning or teaching computer programming

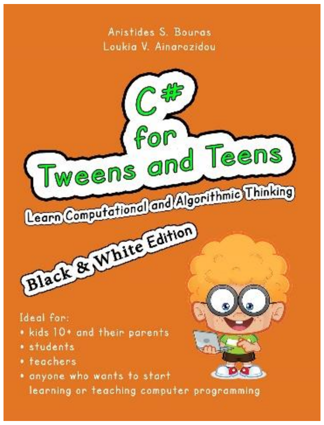
ISBN-10: 1546611215

ISBN-13: 978-1546611219


Aristides S. Bouras
Loukia V. Ainarozidou
C# for Tweens and Teens
Learn Computational and Algorithmic Thinking
Full Color Edition
Ideal for:
• kids 10+ and their parents
• students
• teachers
• anyone who wants to start learning or teaching computer programming

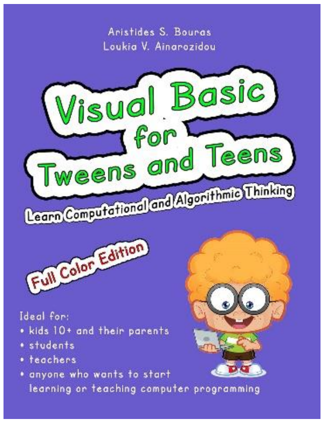
ISBN-10: 1973727684

ISBN-13: 978-1973727682


Aristides S. Bouras
Loukia V. Ainarozidou
C# for Tweens and Teens
Learn Computational and Algorithmic Thinking
Black & White Edition
Ideal for:
• kids 10+ and their parents
• students
• teachers
• anyone who wants to start learning or teaching computer programming

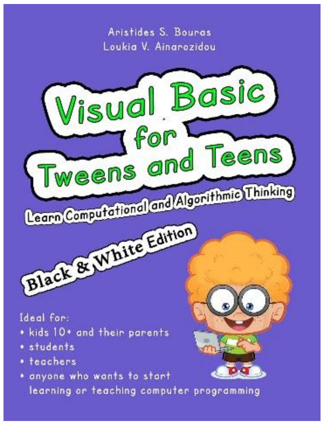
ISBN-10: 1973727765

ISBN-13: 978-1973727767


Aristides S. Bouras
Loukia V. Ainarozidou
Visual Basic for Tweens and Teens
Learn Computational and Algorithmic Thinking
Full Color Edition
Ideal for:
• kids 10+ and their parents
• students
• teachers
• anyone who wants to start learning or teaching computer programming

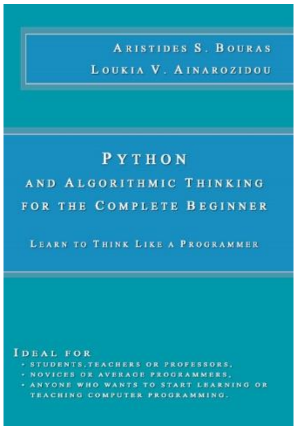
ISBN-10: 1982083670

ISBN-13: 978-1982083670


Aristides S. Bouras
Loukia V. Ainarozidou
Visual Basic for Tweens and Teens
Learn Computational and Algorithmic Thinking
Black & White Edition
Ideal for:
• kids 10+ and their parents
• students
• teachers
• anyone who wants to start learning or teaching computer programming

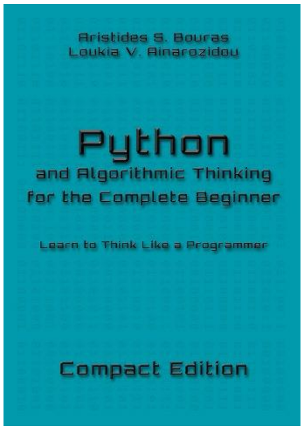
ISBN-10: 1982083697

ISBN-13: 978-1982083694


Aristides S. Bouras
Loukia V. Ainarozidou
Python
and Algorithmic Thinking
for the Complete Beginner
Learn to Think Like a Programmer
Ideal for
• students, teachers or professors,
• novices or average programmers,
• anyone who wants to start learning or teaching computer programming.

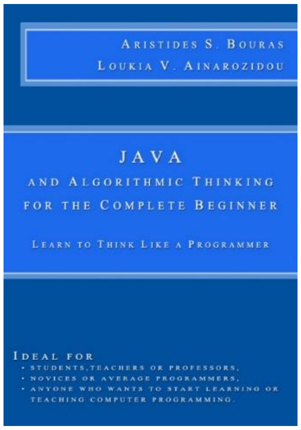
ISBN-10: 1514802163

ISBN-13: 978-1514802168



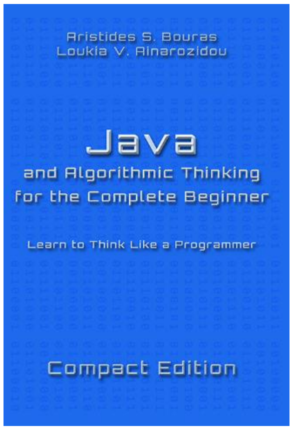
ISBN-10: 1983276111

ISBN-13: 978-1983276118


Aristides S. Bouras
Loukia V. Ainarozidou
Java
and Algorithmic Thinking
for the Complete Beginner
Learn to Think Like a Programmer
Ideal for
• students, teachers or professors,
• novices or average programmers,
• anyone who wants to start learning or teaching computer programming.

ISBN-10: 1506179398

ISBN-13: 978-1506179391



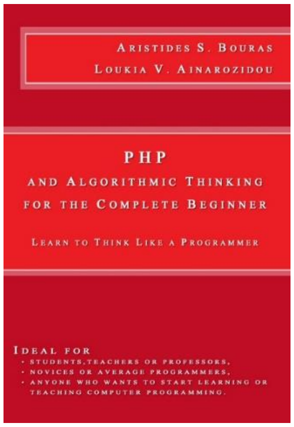
Aristides S. Bouras
Loukia V. Ainarozidou

Java
and Algorithmic Thinking
for the Complete Beginner

Learn to Think Like a Programmer

Compact Edition

ISBN-10: 1723731587
ISBN-13: 978-1723731587



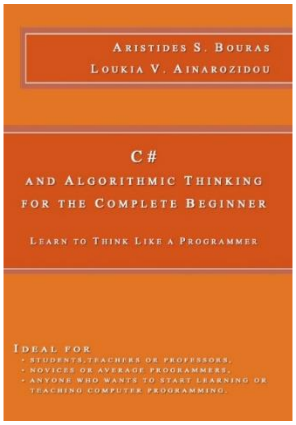
Aristides S. Bouras
Loukia V. Ainarozidou

PHP
and Algorithmic Thinking
for the Complete Beginner

Learn to Think Like a Programmer

Ideal for
- students,teachers or professors,
- novices or average programmers,
- anyone who wants to start learning or teaching computer programming.

ISBN-10: 1503015912
ISBN-13: 978-1503015913



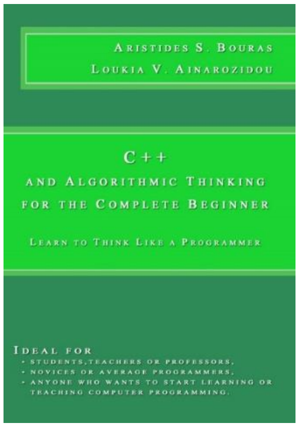
Aristides S. Bouras
Loukia V. Ainarozidou

C#
and Algorithmic Thinking
for the Complete Beginner

Learn to Think Like a Programmer

Ideal for
- students,teachers or professors,
- novices or average programmers,
- anyone who wants to start learning or teaching computer programming.

ISBN-10: 1508952485
ISBN-13: 978-1508952480



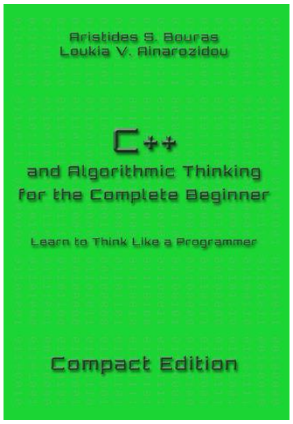
Aristides S. Bouras
Loukia V. Ainarozidou

C++
and Algorithmic Thinking
for the Complete Beginner

Learn to Think Like a Programmer

Ideal for
- students,teachers or professors,
- novices or average programmers,
- anyone who wants to start learning or teaching computer programming.

ISBN-10: 1508577552
ISBN-13: 978-1508577553



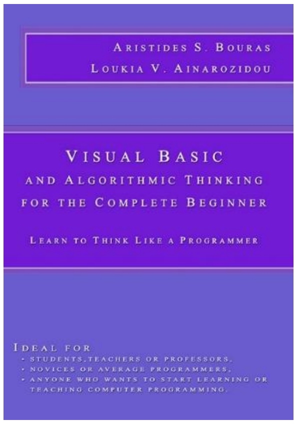
Aristides S. Bouras
Loukia V. Ainarozidou

C++
and Algorithmic Thinking
for the Complete Beginner

Learn to Think Like a Programmer

Compact Edition

ISBN-10: 1726791068
ISBN-13: 978-1726791069



Aristides S. Bouras
Loukia V. Ainarozidou

Visual Basic
and Algorithmic Thinking
for the Complete Beginner

Learn to Think Like a Programmer

Ideal for
- students,teachers or professors,
- novices or average programmers,
- anyone who wants to start learning or teaching computer programming.

ISBN-10: 1511798963
ISBN-13: 978-1511798969