Solutions Companion

C#

AND ALGORITHMIC THINKING FOR THE COMPLETE BEGINNER

3rd Revised Edition

Aristides S. Bouras

https://www.bouraspage.com

RCode: 240301

Warning and Disclaimer

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

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

Table of Contents

How to Report Errata 8
If you Like the Book 8
Chapter 1 9
1.7 Review Questions: True/False 9
1.8 Review Questions: Multiple Choice 9
Review in “Introductory Knowledge” 10
Review Crossword Puzzles 10
Chapter 4 12
4.17 Review Questions: True/False 12
4.18 Review Questions: Multiple Choice 12
Chapter 5 13
5.8 Review Questions: True/False 13
5.9 Review Questions: Multiple Choice 13
5.10 Review Exercises 13
Chapter 6 14
6.4 Review Questions: True/False 14
6.5 Review Questions: Multiple Choice 14
Chapter 7 15
7.7 Review Questions: True/False 15
7.8 Review Questions: Multiple Choice 15
7.9 Review Exercises 15
Chapter 8 16
8.2 Review Questions: True/False 16
8.3 Review Exercises 16
Chapter 9 18
9.5 Review Exercises 18
Review in “Getting Started with C#” 20
Review Crossword Puzzles 20
Chapter 10 21
10.2 Review Exercises 21
Chapter 11 27
11.3 Review Questions: True/False 27
11.4 Review Questions: Multiple Choice 27

11.5 Review Exercises 27
Chapter 12 29
12.2 Review Exercises 29
Chapter 13 31
13.2 Review Exercises 31
Chapter 14 34
14.4 Review Questions: True/False 34
14.5 Review Questions: Multiple Choice 34
14.6 Review Exercises 35
Review in “Sequence Control Structures” 37
Review Crossword Puzzle 37
Chapter 15 38
15.9 Review Questions: True/False 38
15.10 Review Questions: Multiple Choice 38
15.11 Review Exercises 38
Chapter 16 40
16.2 Review Questions: True/False 40
16.3 Review Questions: Multiple Choice 40
16.4 Review Exercises 40
Chapter 17 46
17.2 Review Questions: True/False 46
17.3 Review Questions: Multiple Choice 46
17.4 Review Exercises 46
Chapter 18 54
18.2 Review Questions: True/False 54
18.3 Review Exercises 54
Chapter 19 64
19.2 Review Questions: True/False 64
19.3 Review Exercises 64
Chapter 20 72
20.2 Review Questions: True/False 72
20.3 Review Exercises 72
Chapter 21 80
21.4 Review Exercises 80
Chapter 22 86

22.9 Review Questions: True/False 86
22.10 Review Questions: Multiple Choice 86
22.11 Review Exercises 87
Chapter 23 91
23.7 Review Exercises 91
Review in "Decision Control Structures" 112
Review Crossword Puzzle 112
Chapter 24 113
24.3 Review Questions: True/False 113
Chapter 25 114
25.4 Review Questions: True/False 114
25.5 Review Questions: Multiple Choice 114
25.6 Review Exercises 114
Chapter 26 124
26.3 Review Questions: True/False 124
26.4 Review Questions: Multiple Choice 124
26.5 Review Exercises 125
Chapter 27 140
27.3 Review Questions: True/False 140
27.4 Review Questions: Multiple Choice 140
27.5 Review Exercises 140
Chapter 28 148
28.4 Review Exercises 148
Chapter 29 156
29.8 Review Questions: True/False 156
29.9 Review Questions: Multiple Choice 156
29.10 Review Exercises 156
Chapter 30 161
30.7 Review Questions: True/False 161
30.8 Review Exercises 161
Review in "Loop Control Structures" 184
Review Crossword Puzzle 184
Chapter 31 185
31.13 Review Questions: True/False 185
31.14 Review Questions: Multiple Choice 185

31.15 Review Exercises 185
Chapter 32 199
32.7 Review Questions: True/False 199
32.8 Review Questions: Multiple Choice 199
32.9 Review Exercises 200
Chapter 33 212
33.8 Review Questions: True/False 212
33.9 Review Questions: Multiple Choice 212
33.10 Review Exercises 212
Chapter 34 222
34.7 Review Questions: True/False 222
34.8 Review Exercises 222
Review in “Data Structures in C#” 257
Review Crossword Puzzle 257
Chapter 35 258
35.4 Review Questions: True/False 258
Chapter 36 259
36.8 Review Questions: True/False 259
36.9 Review Exercises 259
Chapter 37 273
37.9 Review Questions: True/False 273
37.10 Review Exercises 273
Chapter 38 282
38.3 Review Exercises 282
Review in “Subprograms” 295
Review Crossword Puzzle 295
Chapter 39 296
39.8 Review Questions: True/False 296
39.9 Review Exercises 296
Review in “Object Oriented Programming” 314
Review Crossword Puzzle 314
Chapter 40 315
40.8 Review Questions: True/False 315
40.9 Review Exercises 315
Chapter 41 318

41.2 Review Exercises 318
Review in “Files” 328
Review Crossword Puzzle 328
Some Final Words from the Author 329
Some of my Books 330

How to Report Errata

Although I have taken great care to ensure the accuracy of the content in this book, mistakes can still occur. If you come across any errors, either in the text or the code, I highly encourage you to report them to me. By doing so, you'll not only assist in saving other readers from potential confusion and frustration but also contribute to enhancing the quality of the next release. If you discover any errors, please report them by visiting one of the following addresses:

- https://tinyurl.com/28nwh2nf
- https://www.bouraspage.com/report-errata

Once I verify your reported error(s), your submission will be accepted. The errata will then be uploaded to my website and added to any existing list of corrections.

If you Like the Book

If you find the book valuable, please consider visiting the web store where you purchased it, as well as goodreads.com, to show your appreciation by writing a positive review and awarding as many stars as you think appropriate. By doing so, you will motivate me to keep writing and, of course, you'll be assisting other readers in discovering my work.

Chapter 1

1.7 Review Questions: True/False

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

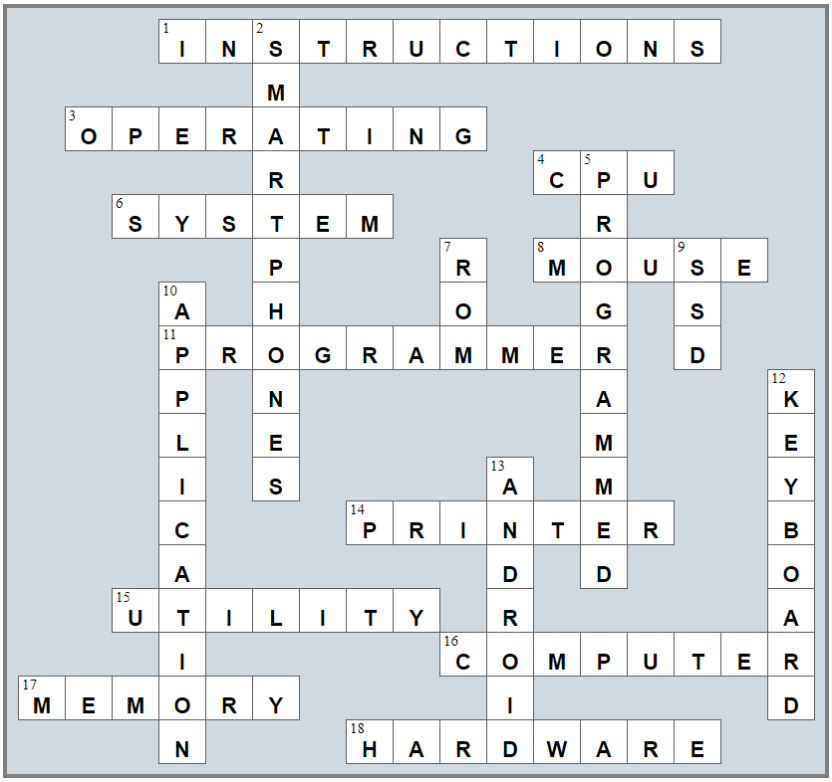
1.8 Review Questions: Multiple Choice

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

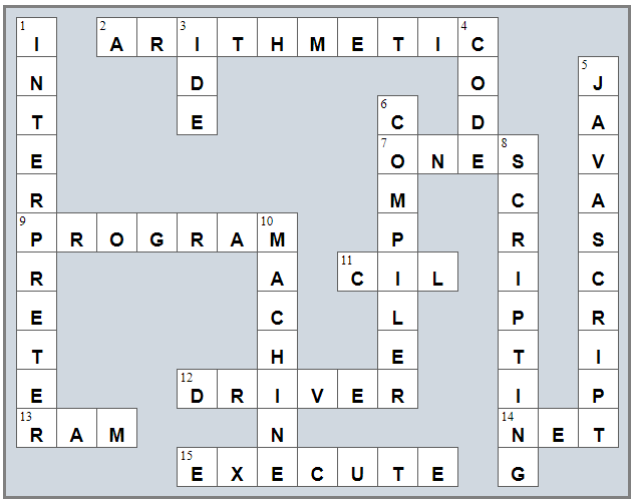
Review in “Introductory Knowledge”

Review Crossword Puzzles

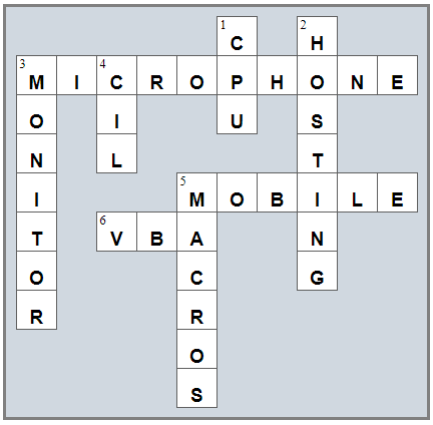
1.



2.



3.



Chapter 4

4.17 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. false
34. true
35. false
36. false
37. false
38. false
39. true
40. true
41. false

4.18 Review Questions: Multiple Choice

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

Chapter 5

5.8 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. true
15. true
16. true
17. false

5.9 Review Questions: Multiple Choice

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

5.10 Review Exercises

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

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

Chapter 6

6.4 Review Questions: True/False

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

6.5 Review Questions: Multiple Choice

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

Chapter 7

7.7 Review Questions: True/False

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

7.8 Review Questions: Multiple Choice

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

7.9 Review Exercises

1. ii, iv, v, ix, x
2. i. String, ii. Boolean, iii. String, iv. String, v. Float, 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. 0, vi. 0
9. i. 2, ii. 5
10. My name is George Malkovich
11. i. (−3), ii. 1
12. California California California

Chapter 8

8.2 Review Questions: True/False

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

8.3 Review Exercises

1. Solution

Step	Statement	Notes	a	b	c	d
1	`a = Convert.ToDouble(Console.ReadLine())`	User enters value 3	**3.0**	?	?	?
2	`b = a + 10`		3.0	**13.0**	?	?
3	`a = b * (a - 3)`		**0.0**	13.0	?	?
4	`c = 3 * b / 6`		0.0	13.0	**6.5**	?
5	`d = c * c`		0.0	13.0	6.5	**42.25**
6	`d--`		0.0	13.0	6.5	**41.25**
7	`Console.WriteLine(d)`	It displays: 41.25				

2. Solution

For the input value of 3

Step	Statement	a	b	c	d
1	`a = Convert.ToInt32(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.WriteLine(a + ", " + b + ", " + c + ", " + d)`	It displays: 40, 1, 1, 40			

For the input value of 4

Step	Statement	a	b	c	d
1	`a = Convert.ToInt32(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.WriteLine(a + ", " + b + ", " + c + ", " + d)`	It displays: 49, 10, 3, 1470			

For the input value of 1

Step	Statement	a	b	c	d
1	`a = Convert.ToInt32(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.WriteLine(a + ", " + b + ", " + c + ", " + d)`	It displays: 28, 2, 2, 112			

3. Solution

For the input values of 8, 4

Step	Statement	a	b	c	d	e
1	`a = Convert.ToInt32(Console.ReadLine())`	**8**	?	?	?	?
2	`b = Convert.ToInt32(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 + d % c`	8	4	78	**–78**	38
9	`Console.WriteLine(c + ", " + d + ", " + e)`	It displays: 78, –78, 38				

For the input values of 4, 4

Step	Statement	a	b	c	d	e
1	`a = Convert.ToInt32(Console.ReadLine())`	**4**	?	?	?	?
2	`b = Convert.ToInt32(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 + d % c`	4	4	38	**–38**	18
9	`Console.WriteLine(c + ", " + d + ", " + e)`	It displays: 38, –38, 18				

Chapter 9

9.5 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 = Convert.ToInt32(Console.ReadLine())`	**5**	?	?	?	?
2	`b = Convert.ToInt32(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.WriteLine(c + ", " + d + ", " + e)`	It displays: 37, 11, –8				

For the input values of 4, 2

Step	Statement	a	b	c	d	e
1	`a = Convert.ToInt32(Console.ReadLine())`	**4**	?	?	?	?
2	`b = Convert.ToInt32(Console.ReadLine())`	4	**2**	?	?	?
3	`c = a + b`	4	2	**6**	?	?
4	`d = 5 + a / b * c + 2`	4	2	6	**19**	?
5	`e = c - d`	4	2	6	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.WriteLine(c + ", " + d + ", " + e)`	It displays: 31, 6, –14				

3. Solution

For the input value of 5

Step	Statement	a	b	c
1	`b = Convert.ToInt32(Console.ReadLine())`	?	**5**	?
2	`c = 5`	?	5	**5**
3	`c = c * b`	?	5	**25**
4	`a = 3 * c % 10`	**5**	5	25

5	`Console.WriteLine(a)`	It displays: 5

For the input value of 4

Step	Statement	a	b	c
1	`b = Convert.ToInt32(Console.ReadLine())`	?	**4**	?
2	`c = 5`	?	4	**5**
3	`c = c * b`	?	4	**20**
4	`a = 3 * c % 10`	**0**	4	20
5	`Console.WriteLine(a)`	It displays: 0		

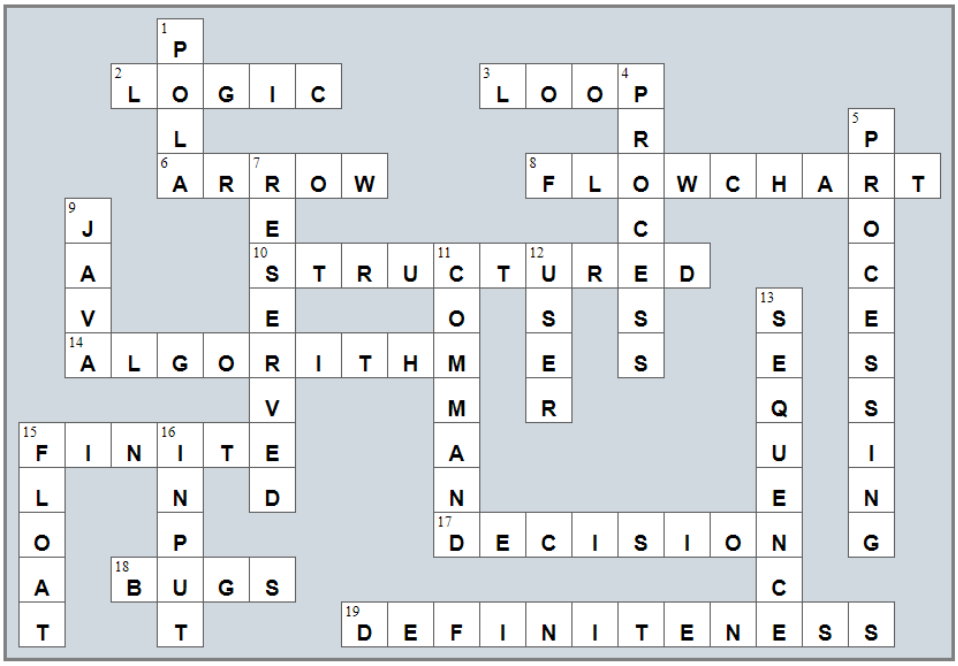
For the input value of 15

Step	Statement	a	b	c
1	`b = Convert.ToInt32(Console.ReadLine())`	?	**15**	?
2	`c = 5`	?	15	**5**
3	`c = c * b`	?	15	**75**
4	`a = 3 * c % 10`	**5**	15	75
5	`Console.WriteLine(a)`	It displays: 5		

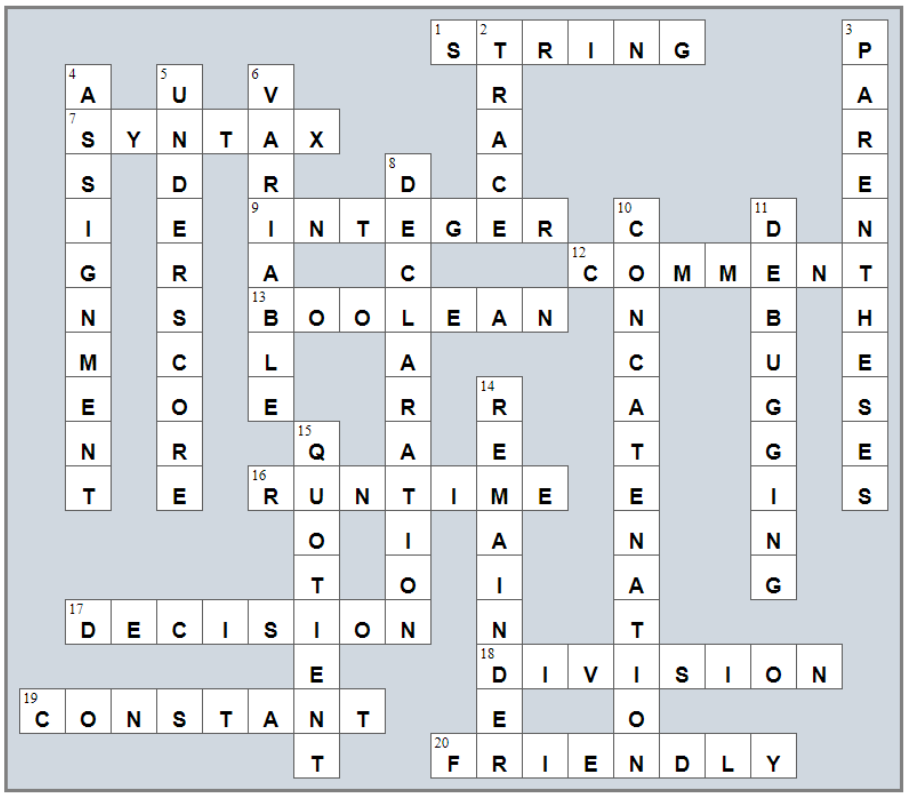
Review in “Getting Started with C#”

Review Crossword Puzzles

1.



2.



Chapter 10

10.2 Review Exercises

1. Solution

```
double gallons, milesDriven, mpg;

Console.Write("Enter miles driven: ");
milesDriven = Convert.ToDouble(Console.ReadLine());
Console.Write("Enter gallons of gas used: ");
gallons = Convert.ToDouble(Console.ReadLine());

mpg = milesDriven / gallons;

Console.WriteLine("Your car's MPG is: " + mpg);
```

2. Solution

```
double b, h, area;

Console.Write("Enter base: ");
b = Convert.ToDouble(Console.ReadLine());
Console.Write("Enter height: ");
h = Convert.ToDouble(Console.ReadLine());

area = 0.5 * b * h;

Console.WriteLine(area);
```

3. Solution

```
double angle1, angle2, angle3;

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

angle3 = 180 - angle1 - angle2;

Console.WriteLine(angle3);
```

4. Solution

```
int g1, g2, g3, g4;
double average;

Console.Write("Enter 1st grade: ");
g1 = Convert.ToDouble(Console.ReadLine());
Console.Write("Enter 2nd grade: ");
g2 = Convert.ToDouble(Console.ReadLine());
Console.Write("Enter 3rd grade: ");
```

```
g3 = Convert.ToDouble(Console.ReadLine());
Console.Write("Enter 4th grade: ");
g4 = Convert.ToDouble(Console.ReadLine());

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

Console.WriteLine(average);
```

5. Solution

```
const double PI = 3.14159;

double r, perimeter;

Console.Write("Enter radius: ");
r = Convert.ToDouble(Console.ReadLine());

perimeter = 2 * PI * r;

Console.WriteLine(perimeter);
```

6. Solution

```
const double PI = 3.14159;

double d, radius, volume;

Console.Write("Enter diameter (in meters): ");
d = Convert.ToDouble(Console.ReadLine());

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

Console.WriteLine(volume);
```

7. Solution

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

8. Solution

```
string firstName, lastName, middleName, title;

Console.Write("First name: ");
firstName = Console.ReadLine();
Console.Write("Middle name: ");
middleName = Console.ReadLine();
Console.Write("Last name: ");
lastName = Console.ReadLine();
Console.Write("Title: ");
title = Console.ReadLine();

Console.WriteLine(title + " " + firstName + " " + middleName + " " + lastName);
Console.WriteLine(firstName + " " + middleName + " " + lastName);
Console.WriteLine(lastName + ", " + firstName);
```

```
Console.WriteLine(lastName + ", " + firstName + " " + middleName);
Console.WriteLine(lastName + ", " + firstName + " " + middleName + ", " + title);
Console.WriteLine(firstName + " " + lastName);
```

9. Solution

```
const double PI = 3.14159;

double d, radius, perimeter, area, volume;

Console.Write("Enter diameter: ");
d = Convert.ToDouble(Console.ReadLine());

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

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

10. Solution

```
double charge, tip, tax, total;

Console.Write("Enter charge for a meal: ");
charge = Convert.ToDouble(Console.ReadLine());

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

total = charge + tip + tax;

Console.WriteLine(total);
```

11. Solution

```
int minutes, seconds, totalSeconds;
double s, a;

Console.Write("Enter the distance traveled (in meters): ");
s = Convert.ToDouble(Console.ReadLine());
Console.Write("Enter the minutes traveled: ");
minutes = Convert.ToInt32(Console.ReadLine());
Console.Write("Enter the seconds traveled: ");
seconds = Convert.ToInt32(Console.ReadLine());

totalSeconds = minutes * 60 + seconds;

a = 2 * s / (totalSeconds * totalSeconds);

Console.WriteLine(a);
```

12. Solution

```
double f, c;
```

```
Console.Write("Enter temperature in Fahrenheit: ");
f = Convert.ToDouble(Console.ReadLine());

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

Console.WriteLine(c);
```

13. Solution

```
int w, h;
double bmi;

Console.Write("Enter weight in pounds: ");
w = Convert.ToInt32(Console.ReadLine());
Console.Write("Enter height in inches: ");
h = Convert.ToInt32(Console.ReadLine());

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

Console.WriteLine(bmi);
```

14. Solution

```
double sTotal, tip, total;
int gRate;

Console.Write("Enter subtotal: ");
sTotal = Convert.ToDouble(Console.ReadLine());
Console.Write("Enter gratuity rate (0 - 100): ");
gRate = Convert.ToInt32(Console.ReadLine());

tip = sTotal * gRate / 100;

total = sTotal + tip;

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

15. Solution

```
const double VAT = 0.20;

double btaxPrice1, btaxPrice2, btaxPrice3, ataxPrice1, ataxPrice2, ataxPrice3, avg;

Console.Write("Enter before-tax price 1: ");
btaxPrice1 = Convert.ToDouble(Console.ReadLine());
Console.Write("Enter before-tax price 2: ");
btaxPrice2 = Convert.ToDouble(Console.ReadLine());
Console.Write("Enter before-tax price 3: ");
btaxPrice3 = Convert.ToDouble(Console.ReadLine());

ataxPrice1 = btaxPrice1 + btaxPrice1 * VAT;
ataxPrice2 = btaxPrice2 + btaxPrice2 * VAT;
ataxPrice3 = btaxPrice3 + btaxPrice3 * VAT;
```

```
avg = (ataxPrice1 + ataxPrice2 + ataxPrice3) / 3;

Console.WriteLine(avg);
```

16. Solution

```
const int VAT = 0.20;

double ataxPrice, btaxPrice;

Console.Write("Enter after-tax price: ");
ataxPrice = Convert.ToInt32(Console.ReadLine());

btaxPrice = ataxPrice / (1 + VAT);

Console.WriteLine(btaxPrice);
```

17. Solution

```
double iPrice, fPrice, saved;
int discount;

Console.Write("Enter price: ");
iPrice = Convert.ToDouble(Console.ReadLine());
Console.Write("Enter discount (0 - 100): ");
discount = Convert.ToInt32(Console.ReadLine());

fPrice = iPrice - iPrice * discount / 100;
saved = iPrice - fPrice;

Console.WriteLine(fPrice + " " + saved);
```

18. Solution

```
const int VAT = 0.20;

int iKWh, fKWh, kWhConsumed;
double cost;

Console.Write("Enter kWh at the beginning of the month: ");
iKWh = Convert.ToInt32(Console.ReadLine());
Console.Write("Enter kWh at the end of the month: ");
fKWh = Convert.ToInt32(Console.ReadLine());

kWhConsumed = fKWh - iKWh;

cost = kWhConsumed * 0.06;
cost += cost * VAT;

Console.WriteLine(kWhConsumed + " " + cost);
```

19. Solution

```
int soldYachts;
double yachtsCost, insuranceCost, totalCost, totalEarnings;
```

```
Console.Write("Enter number of yachts sold: ");
soldYachts = Convert.ToInt32(Console.ReadLine());

yachtsCost = soldYachts * 1000000;
insuranceCost = 250000 * 12;
totalCost = yachtsCost + insuranceCost;
totalEarnings = soldYachts * 1500000;

Console.WriteLine(totalEarnings - totalCost);
```

20. Solution

```
int day, month, daysPassed;

Console.Write("Enter current month: ");
month = Convert.ToInt32(Console.ReadLine());
Console.Write("Enter current day: ");
day = Convert.ToInt32(Console.ReadLine());

daysPassed = (month - 1) * 30 + day;

Console.WriteLine(daysPassed);
```

21. Solution

```
int day, month, daysPassed, daysLeft;

Console.Write("Enter current month: ");
month = Convert.ToInt32(Console.ReadLine());
Console.Write("Enter current day: ");
day = Convert.ToInt32(Console.ReadLine());

daysPassed = (month - 1) * 30 + day;
daysLeft = 360 - daysPassed;

Console.WriteLine(daysLeft);
```

Chapter 11

11.3 Review Questions: True/False

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

11.4 Review Questions: Multiple Choice

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

11.5 Review Exercises

1. Solution

For the input value of 9

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

For the input value of 4

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

2. Solution

For the input value of −2

Step	Statement	a	b	c
1	`a = Convert.ToInt32(Console.ReadLine())`	**−2**	?	?
2	`b = Math.Abs(a) % 4 + Math.Pow(a, 4)`	−2	**18**	**?**
3	`c = b % 5`	−2	18	**3**
4	`Console.WriteLine(b + ", " + c)`	It displays: 18, 3		

For the input value of –3

Step	Statement	a	b	c
1	`a = Convert.ToInt32(Console.ReadLine())`	**–3**	?	?
2	`b = Math.Abs(a) % 4 + Math.Pow(a, 4)`	–3	**84**	**?**
3	`c = b % 5`	–3	84	**4**
4	`Console.WriteLine(b + ", " + c)`	It displays: 84, 4		

3. Solution

```
double degrees, radians;

Console.Write("Enter angle in radians: ");
radians = Convert.ToDouble(Console.ReadLine());

degrees = radians * 180 / Math.PI;

Console.WriteLine(degrees);
```

4. Solution

```
double a, b, hypotenuse;

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

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

Console.WriteLine(hypotenuse);
```

5. Solution

```
double adjacent, opposite, th;

Console.Write("Enter angle θ (in degrees) of a right-angled triangle: ");
th = Convert.ToDouble(Console.ReadLine());
Console.Write("Enter length of adjacent side: ");
adjacent = Convert.ToDouble(Console.ReadLine());

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

Console.WriteLine(opposite);
```

Chapter 12

12.2 Review Exercises

1. Solution

i. a, e, g, h

ii. c, f

2. Solution

i. `y = 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) + 6`

3. Solution

```
double x, y;

Console.Write("Enter value for x: ");
x = Convert.ToDouble(Console.ReadLine());

y = Math.Sqrt(x * x + 1) * (Math.Pow(x, 3) + Math.Pow(x, 2));

Console.WriteLine(y);
```

4. Solution

```
double x, y;

Console.Write("Enter value for x: ");
x = Convert.ToDouble(Console.ReadLine());

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

Console.WriteLine(y);
```

5. Solution

```
double w, x, y;

Console.Write("Enter value for x: ");
x = Convert.ToDouble(Console.ReadLine());
Console.Write("Enter value for w: ");
w = Convert.ToDouble(Console.ReadLine());
```

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

Console.WriteLine(y);
```

6. Solution

```
double w, x, y;

Console.Write("Enter value for x: ");
x = Convert.ToDouble(Console.ReadLine());
Console.Write("Enter value for w: ");
w = Convert.ToDouble(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, 6));

Console.WriteLine(y);
```

7. Solution

```
double w, x, y;

Console.Write("Enter value for x: ");
x = Convert.ToDouble(Console.ReadLine());
Console.Write("Enter value for w: ");
w = Convert.ToDouble(Console.ReadLine());

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

Console.WriteLine(y);
```

8. Solution

```
double a, b, c, area, semi;

Console.Write("Enter length A: ");
a = Convert.ToDouble(Console.ReadLine());
Console.Write("Enter length B: ");
b = Convert.ToDouble(Console.ReadLine());
Console.Write("Enter length C: ");
c = Convert.ToDouble(Console.ReadLine());

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

Console.WriteLine(area);
```

Chapter 13

13.2 Review Exercises

1. Solution

```
int lastDigit, n, result;

Console.Write("Enter an integer: ");
n = Convert.ToInt32(Console.ReadLine());

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

Console.WriteLine(result);
```

2. Solution

```
int digit1, digit2, digit3, digit4, digit5, number, r, reversedNumber;

Console.Write("Enter a five-digit integer: ");
number = Convert.ToInt32(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);

reversedNumber = digit5 * 10000 + digit4 * 1000 + digit3 * 100 + digit2 * 10 + digit1;

Console.WriteLine(number + " + " + reversedNumber + " = " + (number + reversedNumber));
```

3. Solution

```
int n, result;

Console.Write("Enter an integer: ");
n = Convert.ToInt32(Console.ReadLine());

result = n % 2;

Console.WriteLine(result);
```

4. Solution

```
int n, result;
```

```
Console.Write("Enter an integer: ");
n = Convert.ToInt32(Console.ReadLine());

result = 1 - n % 2;

Console.WriteLine(result);
```

5. Solution

```
int days, hours, minutes, number, r, seconds, weeks;

Console.Write("Enter an elapsed time in seconds: ");
number = Convert.ToInt32(Console.ReadLine());

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

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

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

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

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

6. Solution

```
int amount, r, usd1, usd10, usd20, usd5;

Console.Write("Enter amount of money to withdraw: ");
amount = Convert.ToInt32(Console.ReadLine());

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

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

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

Console.Write(usd20 + " note(s) of $20 " + usd10 + " note(s) of $10 ");
Console.WriteLine(usd5 + " note(s) of $5 and " + usd1 + " note(s) of $1");
```

7. Solution

```
int distance, feet, inches, miles, r, steps, yards;

Console.Write("Enter number of steps: ");
steps = Convert.ToInt32(Console.ReadLine());
```

```
distance = steps * 25;

miles = (int)(distance / 63360);
r = distance % 63360;

yards = (int)(r / 36);
r = r % 36;

feet = (int)(r / 12);
inches = r % 12;

Console.Write(miles + " mile(s) " + yards + " yard(s) ");
Console.WriteLine(feet + " foot/feet and " + inches + " inch(es)");
```

Chapter 14

14.4 Review Questions: True/False

1. true
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. true
17. true

14.5 Review Questions: Multiple Choice

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

14.6 Review Exercises

1. Solution

```
string alphabet, randomWord;

Random rnd = new();

alphabet = "abcdefghijklmnopqrstuvwxyz";

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

Console.WriteLine(randomWord);
```

2. Solution

```
string name, x, secretPassword;
Random rnd = new();

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

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

secretPassword = "" + x[rnd.Next(0, x.Length)] +
                      x[rnd.Next(0, x.Length)] +
                      x[rnd.Next(0, x.Length)] +
                      rnd.Next(1000, 10000);

Console.WriteLine(secretPassword);
```

3. Solution

First approach

```
int number, reversedNumber;
string sNumber, digit1, digit2, digit3;

Console.Write("Enter a three-digit integer: ");
number = Convert.ToInt32(Console.ReadLine());

sNumber = "" + number;

digit1 = "" + sNumber[0];
digit2 = "" + sNumber[1];
digit3 = "" + sNumber[2];

reversedNumber = 100 * Convert.ToInt32(digit3) + 10 * Convert.ToInt32(digit2) + Convert.ToInt32(digit1);

Console.WriteLine(reversedNumber);
```

Second approach

```
int number, reversedNumber;
string sNumber;

Console.Write("Enter a three-digit integer: ");
number = Convert.ToInt32(Console.ReadLine());

sNumber = "" + number;
reversedNumber = Convert.ToInt32("" + sNumber[2] + sNumber[1] + sNumber[0]);

Console.WriteLine(reversedNumber);
```

4. Solution

```
string firstName, lastName, middleName;

Console.Write("First name: ");
firstName = Console.ReadLine();
Console.Write("Middle name: ");
middleName = Console.ReadLine();
Console.Write("Last name: ");
lastName = Console.ReadLine();

firstName = firstName.Substring(0, 1).ToUpper() + firstName.Substring(1).ToLower();
middleName = middleName.Substring(0, 1).ToUpper() + middleName.Substring(1).ToLower();
lastName = lastName.Substring(0, 1).ToUpper() + lastName.Substring(1).ToLower();

Console.WriteLine(firstName + " " + middleName + " " + lastName);
Console.WriteLine(firstName + " " + middleName[0] + ". " + lastName);
Console.WriteLine(lastName + " " + firstName[0] + ".");
```

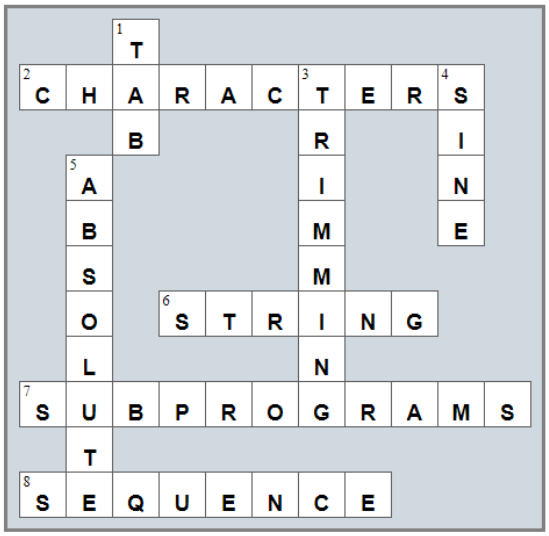
5. Solution

```
string word, abbreviation;

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

abbreviation = "" + word[0] + (word.Length - 2) + word[word.Length - 1];
Console.WriteLine(abbreviation);
```

Review in “Sequence Control Structures”

Review Crossword Puzzle

1.



Chapter 15

15.9 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. true
13. true
14. true
15. false
16. false
17. false
18. true
19. true
20. false
21. true
22. true
23. true

15.10 Review Questions: Multiple Choice

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

15.11 Review Exercises

1. Solution

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

2. Solution

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

3. Solution

BE1 (Boolean Expression 1)	BE2 (Boolean Expression 2)	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 Review Questions: True/False

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

16.3 Review Questions: Multiple Choice

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

16.4 Review Exercises

1. Solution

The corrections/additions are in red

```
double x, y;

x = Convert.ToDouble(Console.ReadLine());

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

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

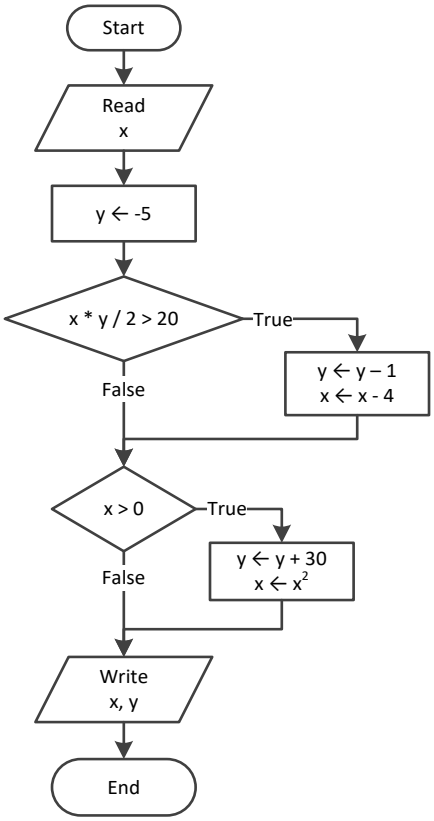
2. Solution

For the input value of 10

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

For the input value of –10

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



3. *Solution*

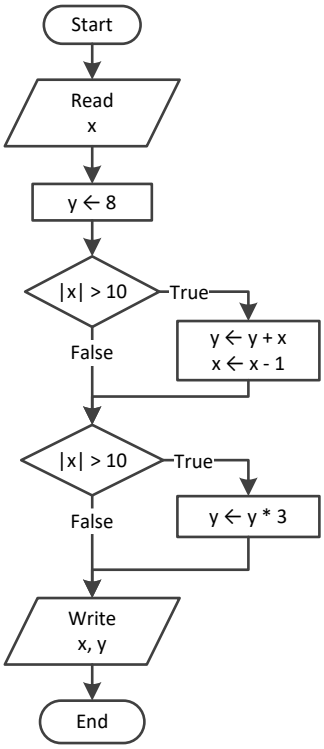
For the input value of –11

Step	Statement	x	y
1	`x = Convert.ToInt32(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.WriteLine(x + ", " + y)`	It displays: –12, –9

For the input value of 11

Step	Statement	x	y
1	`x = Convert.ToInt32(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.WriteLine(x + ", " + y)`	It displays: 10, 19	



4. *Solution*

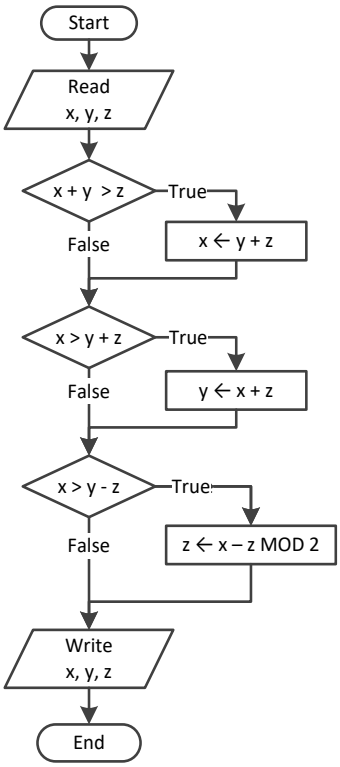
For input values of 1, 2 and 3

Step	Statement	x	y	z
1	`x = Convert.ToInt32(Console.ReadLine())`	**1**	?	?
2	`y = Convert.ToInt32(Console.ReadLine())`	1	**2**	?
3	`z = Convert.ToInt32(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.WriteLine(x + ", " + y + ", " + z)`	It displays: 1, 2, 0

For input values of 4, 2 and 1

Step	Statement	x	y	z
1	`x = Convert.ToInt32(Console.ReadLine())`	**4**	?	?
2	`y = Convert.ToInt32(Console.ReadLine())`	4	**2**	?
3	`z = Convert.ToInt32(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.WriteLine(x + ", " + y + ", " + z)`	It displays: 3, 2, 2		



5. Solution

```
double x;

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

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

6. Solution

```
double x, y;

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

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

7. Solution

```
double x, y;

Console.Write("Enter your age: ");
x = Convert.ToDouble(Console.ReadLine());

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

8. Solution

```
string str;

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

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

9. Solution

```
string str;

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

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

10. Solution

```
double n1, n2, n3, n4;

Console.Write("Enter 1st number: ");
n1 = Convert.ToDouble(Console.ReadLine());
Console.Write("Enter 2nd number: ");
n2 = Convert.ToDouble(Console.ReadLine());
```

```
Console.Write("Enter 3rd number: ");
n3 = Convert.ToDouble(Console.ReadLine());
Console.Write("Enter 4th number: ");
n4 = Convert.ToDouble(Console.ReadLine());

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

11. Solution

```
double a, b, c;

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

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

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

12. Solution

```
double average, t1, t2, t3;

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

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

if (average > 60) {
  Console.WriteLine("Heat Wave");
}
```

Chapter 17

17.2 Review Questions: True/False

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

17.3 Review Questions: Multiple Choice

1. b
2. c
3. c

17.4 Review Exercises

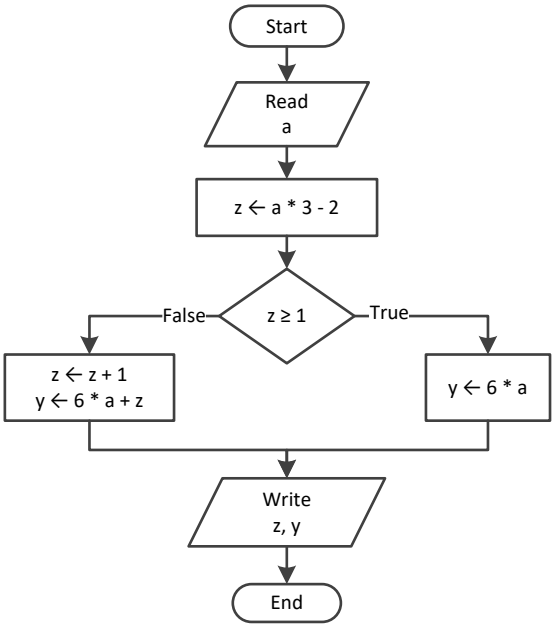
1. Solution

For input value of 3

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

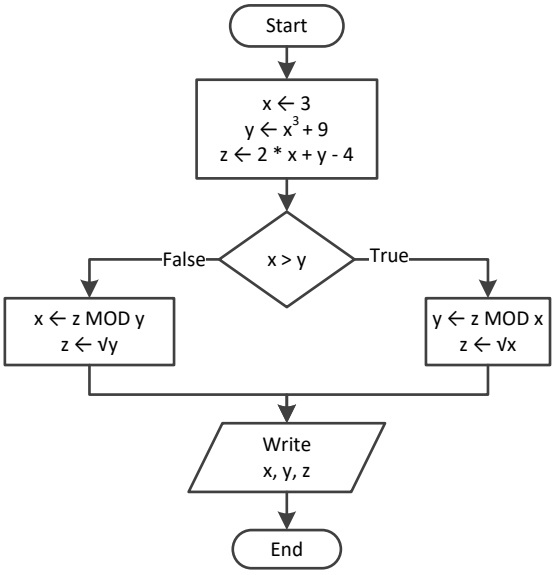
For input value of 0.5

Step	Statement	a	y	z
1	`a = Convert.ToDouble(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.WriteLine(z + ", " + y)`	It displays: 0.5, 3.5		



2. Solution

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



3. Solution

```
double w, x, y, z;

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

For input value of 10

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

For input value of 2

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

4. Solution

```
double num;

Console.Write("Enter a number: ");
num = Convert.ToDouble(Console.ReadLine());

if (num > 100) {
  Console.WriteLine("Provided number is greater than 100");
}
else {
  Console.WriteLine("Provided number is less than or equal to 100");
```

```
}
```

5. Solution

```
double num;

Console.Write("Enter a number: ");
num = Convert.ToDouble(Console.ReadLine());

if (num >= 0 && num <= 100) {
  Console.WriteLine("Provided number is between 0 and 100");
}
else {
  Console.WriteLine("Provided number is not between 0 and 100");
}
```

6. Solution

```
string name1, name2;
int goals1, goals2;

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

Console.Write("Enter goals " + name1 + " scored: ");
goals1 = Convert.ToInt32(Console.ReadLine());
Console.Write("Enter goals " + name2 + " scored: ");
goals2 = Convert.ToInt32(Console.ReadLine());

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

7. Solution

```
int x;

x = Convert.ToInt32(Console.ReadLine());
if (x % 6 == 0) {
  Console.WriteLine(x + " is a multiple of 6");
}
else {
  Console.WriteLine(x + " is not a multiple of 6");
}
```

8. Solution

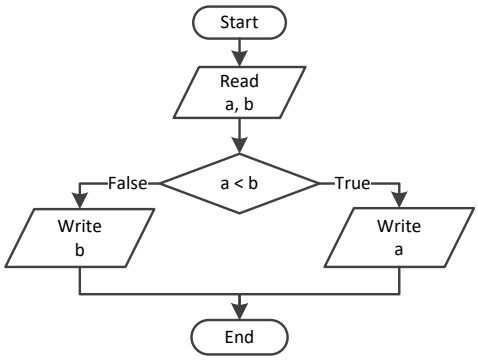
```
int x;
```

```
x = Convert.ToInt32(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");
}
```

9. Solution

```
int x, y;

x = Convert.ToInt32(Console.ReadLine());

y = x % 4;
if (y == 0) {
  Console.WriteLine(x + " is a multiple of 4");
}
else {
  Console.WriteLine(x + " is not a multiple of 4");
}

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

10. Solution




```
double a, b;

a = Convert.ToDouble(Console.ReadLine());
b = Convert.ToDouble(Console.ReadLine());

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

11. Solution

```
double a, b, c;
```

```
a = Convert.ToDouble(Console.ReadLine());
b = Convert.ToDouble(Console.ReadLine());
c = Convert.ToDouble(Console.ReadLine());

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

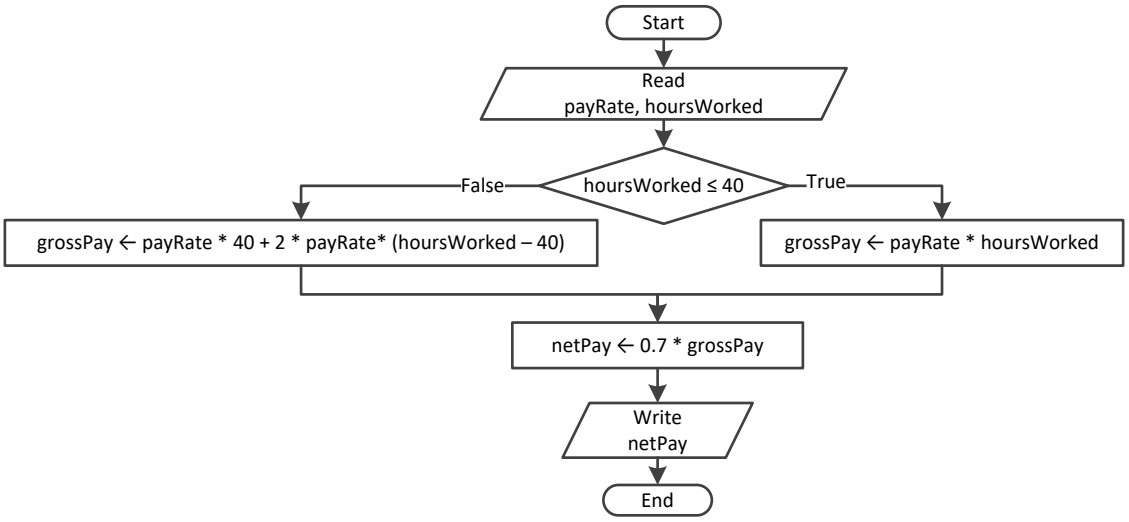
12. Solution

```
double a, b, c;

a = Convert.ToDouble(Console.ReadLine());
b = Convert.ToDouble(Console.ReadLine());
c = Convert.ToDouble(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("Provided numbers can be lengths of the three sides of a right triangle");
}
else {
  Console.WriteLine("Provided numbers cannot be lengths of the three sides of a right triangle");
}
```

13. Solution

```
double a, average, b, c;

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

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

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

14. Solution




```
double grossPay, netPay, payRate;
int hoursWorked;

payRate = Convert.ToDouble(Console.ReadLine());
hoursWorked = Convert.ToInt32(Console.ReadLine());

if (hoursWorked <= 40) {
  grossPay = payRate * hoursWorked;
}
else {
  grossPay = payRate * 40 + 2 * payRate * (hoursWorked - 40);
}

netPay = 0.7 * grossPay;
Console.WriteLine(netPay);
```

15. Solution

```
int miles, milesLeft, r;

Console.Write("Enter miles traveled: ");
miles = Convert.ToInt32(Console.ReadLine());

r = miles % 12000;

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

16. Solution

```
double a1, a2, s1, s2, t;

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

s1 = 0.5 * a1 * Math.Pow(t, 2);
s2 = 0.5 * a2 * Math.Pow(t, 2);

Console.WriteLine("Distance between them: " + Math.Abs(s1 - s2) + " meters");

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

Chapter 18

18.2 Review Questions: True/False

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

18.3 Review Exercises

1. Solution

For input value of 5

Step	Statement	q	b
1	`q = Convert.ToInt32(Console.ReadLine())`	**5**	?
2	`if (q > 0 && q <= 50)`	true	
3	`b = 1`	5	**1**
4	`Console.WriteLine(b)`	It displays: 1	

For input value of 150

Step	Statement	q	b
1	`q = Convert.ToInt32(Console.ReadLine())`	**150**	?
2	`if (q > 0 && q <= 50)`	false	
3	`else if (q > 50 && q <= 100)`	false	
4	`else if (q > 100 && q <= 200)`	true	
5	`b = 3`	150	**3**
6	`Console.WriteLine(b)`	It displays: 3	

For input value of 250

Step	Statement	q	b
1	`q = Convert.ToInt32(Console.ReadLine())`	**250**	?
2	`if (q > 0 && q <= 50)`	false	
3	`else if (q > 50 && q <= 100)`	false	
4	`else if (q > 100 && q <= 200)`	false	
5	`b = 4`	250	**4**
6	`Console.WriteLine(b)`	It displays: 4	

For input value of −1

Step	Statement	q	b
1	`q = Convert.ToInt32(Console.ReadLine())`	**−1**	?
2	`if (q > 0 && q <= 50)`	false	
3	`else if (q > 50 && q <= 100)`	false	
4	`else if (q > 100 && q <= 200)`	false	

5	`b = 4`	−1	**4**
6	`Console.WriteLine(b)`	It displays: 4	

2. Solution

For input value of 5

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

For input value of 150

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

For input value of −1

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

3. Solution

```
double a, y;

a = Convert.ToDouble(Console.ReadLine());

if (a < 1) {
  y = 5 + a;
```

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

4. Solution

```
int n1, n2;

Console.Write("Enter an integer: ");
n1 = Convert.ToInt32(Console.ReadLine());
Console.Write("Enter a second integer: ");
n2 = Convert.ToInt32(Console.ReadLine());

if (n1 % 2 == 0 && n2 % 2 == 0) {
  Console.WriteLine("Both numbers are evens");
}
else if (n1 % 2 != 0 && n2 % 2 != 0) {
  Console.WriteLine("Both numbers are odds");
}
else {
  Console.WriteLine("Nothing special!");
}
```

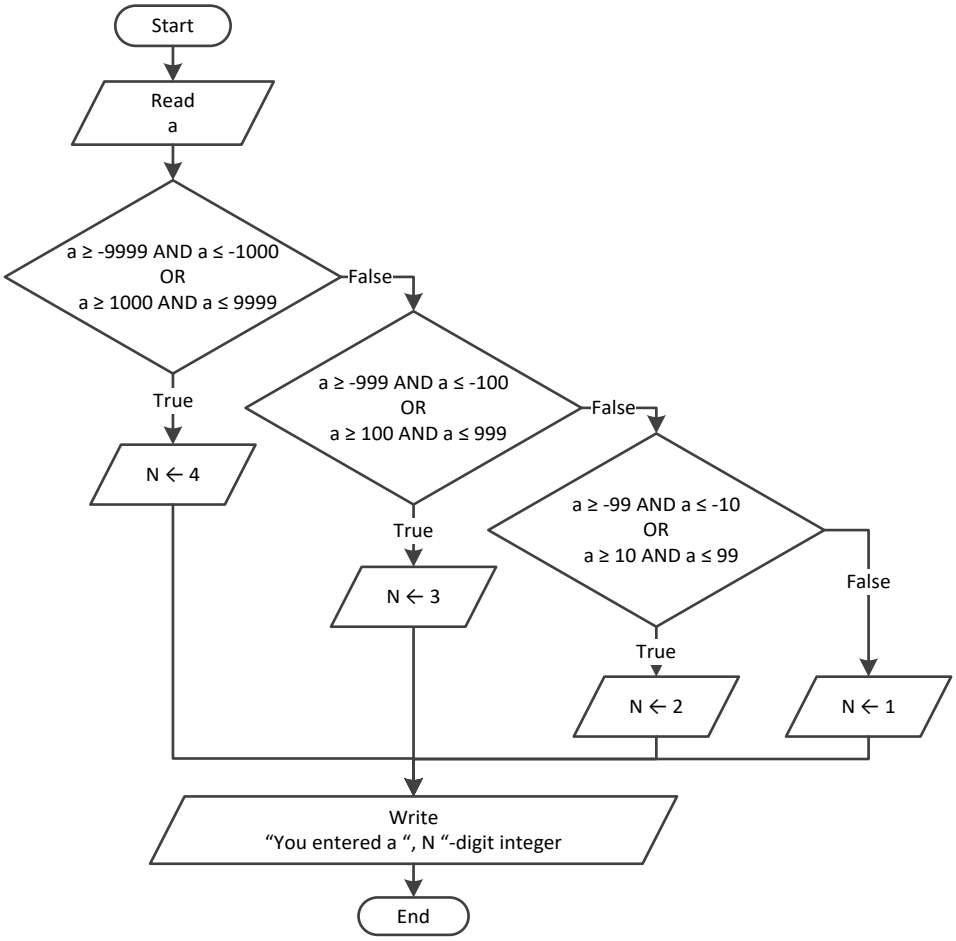
5. Solution

```
string name1, name2;
int goals1, goals2;

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

Console.Write("Enter goals " + name1 + " scored: ");
goals1 = Convert.ToInt32(Console.ReadLine());
Console.Write("Enter goals " + name2 + " scored: ");
goals2 = Convert.ToInt32(Console.ReadLine());

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

```
}
else {
  Console.WriteLine("It's a tie!");
}
```

6. Solution

First approach




```
int a, n;

a = Convert.ToInt32(Console.ReadLine());

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

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

Second approach

```
int a, n;

a = Math.Abs(Convert.ToInt32(Console.ReadLine()));

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

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

Third approach

```
int a;
string aString;

a = Convert.ToInt32(Console.ReadLine());

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

7. Solution

First approach

```
int a, n;

a = Convert.ToInt32(Console.ReadLine());

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

Second approach

```
int a, n;

a = Math.Abs(Convert.ToInt32(Console.ReadLine()));

if (a >= 1000 && a <= 9999) {
  Console.WriteLine("You entered a 4-digit integer");
}
else if (a >= 100 && a <= 999) {
  Console.WriteLine("You entered a 3-digit integer");
}
else if (a >= 10 && a <= 99) {
  Console.WriteLine("You entered a 2-digit integer");
}
else if (a >= 0 && a <= 9) {
  Console.WriteLine("You entered a 1-digit integer");
}
else {
  Console.WriteLine("Error: Invalid value!");
}
```

Third approach

```
int a;
string aString;

a = Convert.ToInt32(Console.ReadLine());

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

8. Solution

```
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 = Convert.ToInt32(Console.ReadLine());

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

if (ch == 1) {
  eur = usd * 0.94;
  Console.WriteLine("$" + usd + " = " + eur + " EUR");
```

```
}
else if (ch == 2) {
  gbp = usd * 0.81;
  Console.WriteLine("$" + usd + " = " + gbp + " GBP");
}
else if (ch == 3) {
  jpy = usd * 149.11;
  Console.WriteLine("$" + usd + " = " + jpy + " JPY");
}
else {
  cad = usd * 1.36;
  Console.WriteLine("$" + usd + " = " + cad + " CAD");
}
```

9. Solution

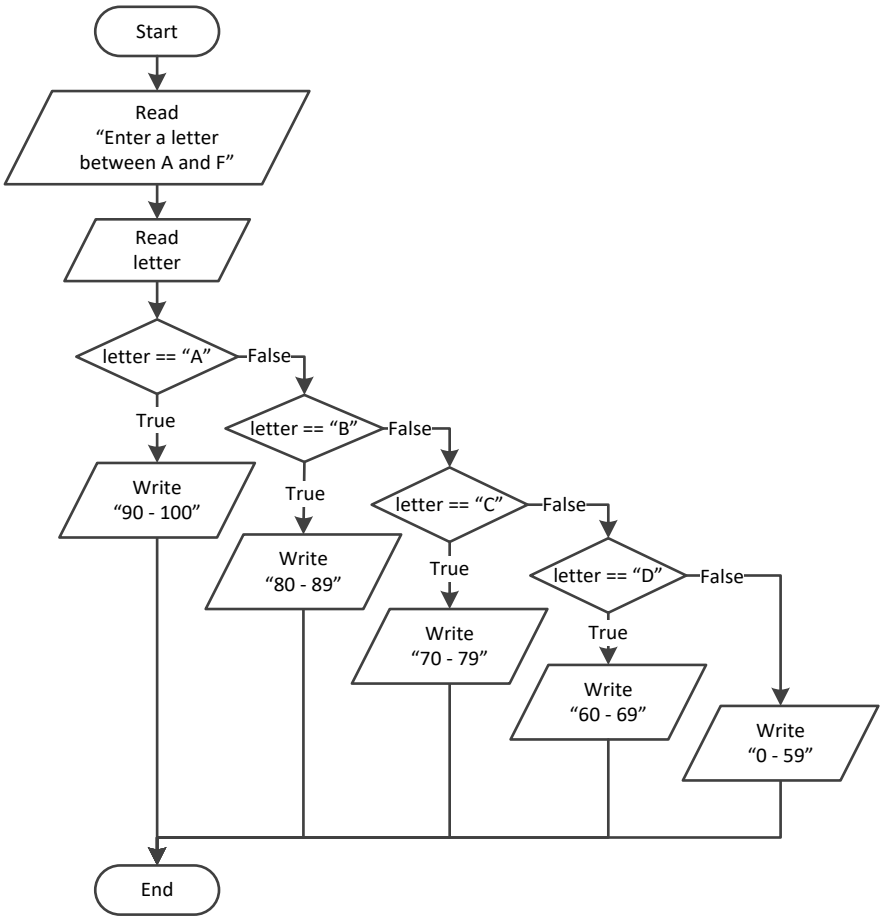
```
int m;

Console.Write("Enter the number of a month between 1 and 12: ");
m = Convert.ToInt32(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)");
}
```

10. Solution

```
int m;

Console.Write("Enter the number of a month between 1 and 12: ");
m = Convert.ToInt32(Console.ReadLine());

if (m < 1 || m > 12) {
  Console.WriteLine("Error: Invalid value!");
else if (m <= 2 || m == 12) {
  Console.WriteLine("Winter");
}
else if (m <= 5) {
  Console.WriteLine("Spring");
}
else if (m <= 8) {
  Console.WriteLine("Summer");
}
```

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

11. Solution




```
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");
```

```
}
```

12. Solution

```
double n;
int x, y;
string number;

Console.Write("Enter a number between 0.0 and 9.9: ");
n = Convert.ToDouble(Console.ReadLine());

x = (int)(n);
y = (int)(n * 10) % 10;

number = "";

if (x == 1) {
  number += "One";
}
else if (x == 2) {
  number += "Two";
}
else if (x == 3) {
  number += "Three";
}
else if (x == 4) {
  number += "Four";
}
else if (x == 5) {
  number += "five";
}
else if (x == 6) {
  number += "six";
}
else if (x == 7) {
  number += "seven";
}
else if (x == 8) {
  number += "eight";
}
else if (x == 9) {
  number += "nine";
}
else if (x == 0) {
  number += "zero";
}

number += " point ";

if (y == 1) {
  number += "one";
}
else if (y == 2) {
```

```
  number += "two";
}
else if (y == 3) {
  number += "three";
}
else if (y == 4) {
  number += "four";
}
else if (y == 5) {
  number += "five";
}
else if (y == 6) {
  number += "six";
}
else if (y == 7) {
  number += "seven";
}
else if (y == 8) {
  number += "eight";
}
else if (y == 9) {
  number += "nine";
}
else if (y == 0) {
  number += "zero";
}

Console.WriteLine(number);
```

Chapter 19

19.2 Review Questions: True/False

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

19.3 Review Exercises

1. Solution

For input value of 1

Step	Statement	a	x	y
1	`a = Convert.ToInt32(Console.ReadLine())`	**1**	?	?
2	`x = 0`	1	**0**	?
3	`y = 0`	1	0	**0**
4	`case a == 1`		true	
5	`x = x + 5`	1	**5**	0
6	`y = y + 5`	1	5	**5**
7	`Console.WriteLine(x + ", " + y)`	It displays: 5, 5		

For input value of 3

Step	Statement	a	x	y
1	`a = Convert.ToInt32(Console.ReadLine())`	**3**	?	?
2	`x = 0`	3	**0**	?
3	`y = 0`	3	0	**0**
4	`case a == 1`		false	
5	`case a == 2`		false	
6	`case a == 3`		true	
7	`x = x - 9`	3	**–9**	0
8	`y = y + 3`	3	–9	**3**
9	`Console.WriteLine(x + ", " + y)`	It displays: –9, 3		

For input value of 250

Step	Statement	a	x	y
1	`a = Convert.ToInt32(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.WriteLine(x + ", " + y)`	It displays: 3, 1		

2. Solution

For input values of 10, 2, 5

Step	Statement	a	x	y
1	`a = Convert.ToInt32(Console.ReadLine())`	**10**	?	?
2	`x = Convert.ToInt32(Console.ReadLine())`	10	**2**	?
3	`y = Convert.ToDouble(Console.ReadLine())`	10	2	**5.0**
4	`case a == 10`	true		
5	`x = x % 2`	10	**0**	5.0
6	`y = Math.Pow(y, 2)`	10	0	**25.0**
7	`Console.WriteLine(x + ", " + y)`	It displays: 0, 25		

For input values of 5, 2, 3

Step	Statement	a	x	y
1	`a = Convert.ToInt32(Console.ReadLine())`	**5**	?	?
2	`x = Convert.ToInt32(Console.ReadLine())`	5	**2**	?
3	`y = Convert.ToDouble(Console.ReadLine())`	5	2	**3.0**
4	`case a == 10`	false		
5	`case a == 3`	false		
6	`case a == 5`	true		
7	`x = x + 4`	5	**6**	3.0
8	`y += 7`	5	6	**10.0**
9	`Console.WriteLine(x + ", " + y)`	It displays: 6, 10		

For input values of 4, 6, 2

Step	Statement	a	x	y
1	`a = Convert.ToInt32(Console.ReadLine())`	**4**	?	?
2	`x = Convert.ToInt32(Console.ReadLine())`	4	**6**	?
3	`y = Convert.ToDouble(Console.ReadLine())`	4	6	**2.0**
4	`case a == 10`	false		
5	`case a == 3`	false		
6	`case a == 5`	false		
7	`x -= 3`	4	**3**	2.0
8	`y++`	4	3	**3.0**
9	`Console.WriteLine(x + ", " + y)`	It displays: 3, 3		

3. Solution

```
string name;

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

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

4. Solution

```
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 = Convert.ToInt32(Console.ReadLine());

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

5. Solution

```
string roman;

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

switch (roman) {
  case "I":
    Console.WriteLine("1");
    break;
  case "II":
    Console.WriteLine("2");
    break;
  case "III":
    Console.WriteLine("3");
```

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

6. Solution

```
int bottles;

Console.Write("Enter the total number of wine bottles purchased in a month: ");
bottles = Convert.ToInt32(Console.ReadLine());

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

7. Solution

```
int i;
```

```
string name;

Random rnd = new();

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

i = rnd.Next(0, 3);

switch (i) {
  case 0:
    Console.WriteLine("Hello " + name + "!");
    break;
  case 1:
    Console.WriteLine("Hi " + name + "!");
    break;
  case 2:
    Console.WriteLine("What's up " + name + "!");
    break;
}
```

8. Solution

```
string num;

num = Console.ReadLine();

switch (num) {
  case "zero":
    Console.WriteLine(0);
    break;
  case "one":
    Console.WriteLine(1);
    break;
  case "two":
    Console.WriteLine(2);
    break;
  case "three":
    Console.WriteLine(3);
    break;
  case "four":
    Console.WriteLine(4);
    break;
  case "five":
    Console.WriteLine(5);
    break;
  case "six":
    Console.WriteLine(6);
    break;
  case "seven":
    Console.WriteLine(7);
    break;
  case "eight":
```

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

9. Solution

```
int b;

Console.Write("Enter Beaufort number: ");
b = Convert.ToInt32(Console.ReadLine());

switch (b) {
  case 0:
    Console.WriteLine("Calm");
    break;
  case 1:
    Console.WriteLine("Light Air");
    break;
  case 2:
    Console.WriteLine("Light breeze");
    break;
  case 3:
    Console.WriteLine("Gentle breeze");
    break;
  case 4:
    Console.WriteLine("Moderate breeze");
    break;
  case 5:
    Console.WriteLine("Fresh breeze");
    break;
  case 6:
    Console.WriteLine("Strong breeze");
    break;
  case 7:
    Console.WriteLine("Moderate gale");
    break;
  case 8:
    Console.WriteLine("Gale");
    break;
  case 9:
    Console.WriteLine("Strong gale");
    break;
  case 10:
    Console.WriteLine("Storm");
    break;
  case 11:
```

```
    Console.WriteLine("Violent storm");
    break;
  case 12:
    Console.WriteLine("Hurricane force");
    break;
  default:
    Console.WriteLine("Invalid Beaufort number!");
    break;
}
```

Chapter 20

20.2 Review Questions: True/False

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

20.3 Review Exercises

1. Solution

For input values of 20, 1

Step	Statement	x	y
1	`x = Convert.ToInt32(Console.ReadLine())`	**20**	?
2	`y = Convert.ToInt32(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.WriteLine(x + ", " + y)`	It displays: 2, 5	

For input values of 20, 3

Step	Statement	x	y
1	`x = Convert.ToInt32(Console.ReadLine())`	**20**	?
2	`y = Convert.ToInt32(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.WriteLine(x + ", " + y)`	It displays: 25, 6	

For input values of 12, 8

Step	Statement	x	y
1	`x = Convert.ToInt32(Console.ReadLine())`	**12**	?
2	`y = Convert.ToInt32(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.WriteLine(x + ", " + y)`	It displays: 10, 9	

For input values of 50, 0

Step	Statement	x	y
1	`x = Convert.ToInt32(Console.ReadLine())`	**50**	?
2	`y = Convert.ToInt32(Console.ReadLine())`	50	**0**
3	`if (x < 30)`	false	
4	`y++`	50	**1**
5	`Console.WriteLine(x + ", " + y)`	It displays: 50, 1	

2. Solution

For input values of 60, 25

Step	Statement	x	y
1	`x = Convert.ToInt32(Console.ReadLine())`	**60**	?
2	`y = Convert.ToInt32(Console.ReadLine())`	60	25
3	`if ((x + y) / 2 <= 20)`	false	
4	`if (y < 15)`	false	
5	`else if (y < 23)`	false	
6	`x = 2 * x + 5`	**125**	25
7	`y += 1`	125	**26**
8	`Console.WriteLine(x + ", " + y)`	It displays: 125, 26	

For input values of 50, 8

Step	Statement	x	y
1	`x = Convert.ToInt32(Console.ReadLine())`	**50**	?
2	`y = Convert.ToInt32(Console.ReadLine())`	50	8
3	`if ((x + y) / 2 <= 20)`	false	
4	`if (y < 15)`	true	
5	`x = x % 4`	**2**	8
6	`y = 2`	2	**2**
7	`Console.WriteLine(x + ", " + y)`	It displays: 2, 2	

For input values of 20, 15

Step	Statement	x	y
1	`x = Convert.ToInt32(Console.ReadLine())`	**20**	?
2	`y = Convert.ToInt32(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.WriteLine(x + ", " + y)`	It displays: 100, 17	

For input values of 10, 30

Step	Statement	x	y
1	`x = Convert.ToInt32(Console.ReadLine())`	**10**	?
2	`y = Convert.ToInt32(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.WriteLine(x + ", " + y)`	It displays: 8, 33	

3. Solution

```
int a;

a = Convert.ToInt32(Console.ReadLine());

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

4. Solution

First approach

```
int age;

Console.Write("Enter your age: ");
age = Convert.ToInt32(Console.ReadLine());

if (age < 0) {
  Console.WriteLine("Error: Invalid age!");
```

```
}
else {
  if (age < 16) {
    Console.WriteLine("You cannot drive either a small scooter or a car");
  }
  else {
    if (age < 18) {
      Console.WriteLine("You can drive a small scooter");
    }
    else {
      Console.WriteLine("You can drive a car and a small scooter");
    }
  }
}
```

Second approach

```
int age;

Console.Write("Enter your age: ");
age = Convert.ToInt32(Console.ReadLine());

if (age < 0) {
  Console.WriteLine("Error: Invalid age!");
}
else {
  if (age < 16) {
    Console.WriteLine("You cannot drive either a small scooter or a car");
  }
  else if (age < 18) {
    Console.WriteLine("You can drive a small scooter");
  }
  else {
    Console.WriteLine("You can drive a car and a small scooter");
  }
}
```

Third approach

```
int age;

Console.Write("Enter your age: ");
age = Convert.ToInt32(Console.ReadLine());

if (age < 0) {
  Console.WriteLine("Error: Invalid age!");
}
else if (age < 16) {
  Console.WriteLine("You cannot drive either a small scooter or a car");
}
else if (age < 18) {
  Console.WriteLine("You can drive a small scooter");
}
else {
  Console.WriteLine("You can drive a car and a small scooter");
```

```
}
```

5. Solution

```
int soldHoverboards, employeesNum;
double hoverboardsCost, insuranceCost, totalCost;
double totalEarnings, profitLoss;

Console.Write("Enter number of hoverboards sold: ");
soldHoverboards = Convert.ToInt32(Console.ReadLine());
Console.Write("Enter number of employees: ");
employeesNum = Convert.ToInt32(Console.ReadLine());

if (soldHoverboards < 0 || employeesNum <= 0) {
  Console.WriteLine("Wrong value(s) entered");
}
else {
  hoverboardsCost = soldHoverboards * 150;
  insuranceCost = employeesNum * 1000;
  totalCost = hoverboardsCost + insuranceCost;

  totalEarnings = soldHoverboards * 250;
  profitLoss = totalEarnings - totalCost;

  if (profitLoss > 0) {
    Console.WriteLine("Profit");
  }
  else if (profitLoss < 0) {
    Console.WriteLine("Loss");
  }
  else {
    Console.WriteLine("Broke even");
  }
}
```

6. Solution

First approach: Using nested decision structures

```
int hour;
string name;

Random rnd = new();

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

hour = rnd.Next(1, 25);
Console.WriteLine("The hour is " + hour + ":00");

if (hour >= 5 && hour <= 11) {
  Console.WriteLine("Good Morning " + name + "!");
}
else {
```

```
  if (hour >= 12 && hour <= 18) {
    Console.WriteLine("Good Afternoon " + name + "!");
  }
  else {
    if (hour >= 19 && hour <= 22) {
      Console.WriteLine("Good Evening " + name + "!");
    }
    else {
      Console.WriteLine("Good Night " + name + "!");
    }
  }
}
```

Second approach: Using a multiple-alternative decision structure

```
int hour;
string name;

Random rnd = new();

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

hour = rnd.Next(1, 25);
Console.WriteLine("The hour is " + hour + ":00");

if (hour >= 5 && hour <= 11) {
  Console.WriteLine("Good Morning " + name + "!");
}
else if (hour >= 12 && hour <= 18) {
  Console.WriteLine("Good Afternoon " + name + "!");
}
else if (hour >= 19 && hour <= 22) {
  Console.WriteLine("Good Evening " + name + "!");
}
else {
  Console.WriteLine("Good Night " + name + "!");
}
```

7. Solution

```
double a, b, c;

Console.Write("Enter the three sides of a triangle: ");
a = Convert.ToDouble(Console.ReadLine());
b = Convert.ToDouble(Console.ReadLine());
c = Convert.ToDouble(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");
  }
}
```

8. Solution

```
int amount, pin, r, usd1, usd10, usd5;

Console.Write("Enter your four-digit PIN : ");
pin = Convert.ToInt32(Console.ReadLine());
if (pin != 1234) {
  Console.Write("Wrong PIN. Enter your four-digit PIN : ");
  pin = Convert.ToInt32(Console.ReadLine());
  if (pin != 1234) {
    Console.Write("Wrong PIN. Enter your four-digit PIN : ");
    pin = Convert.ToInt32(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 = Convert.ToInt32(Console.ReadLine());
  usd10 = (int)(amount / 10);
  r = amount % 10;
  usd5 = (int)(r / 5);
  usd1 = r % 5;
  Console.Write(usd10 + " note(s) of $10 " + usd5 + " note(s) of $5 ");
  Console.WriteLine("and " + usd1 + " note(s) of $1");
}
```

9. Solution

First approach

```
double t, w;

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

if (t > 75) {
  if (w > 12) {
```

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

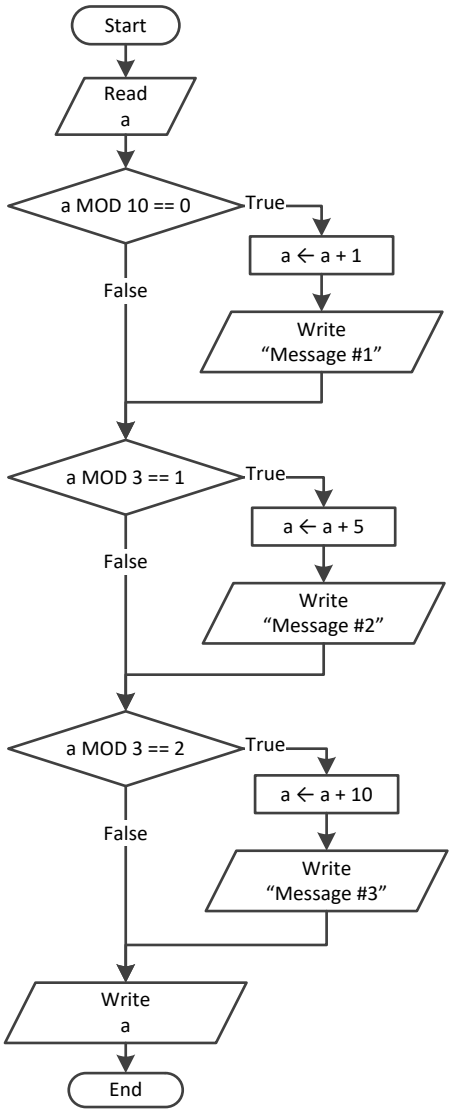
Second approach

```
double t, w;
string message1, message2;

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

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

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

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

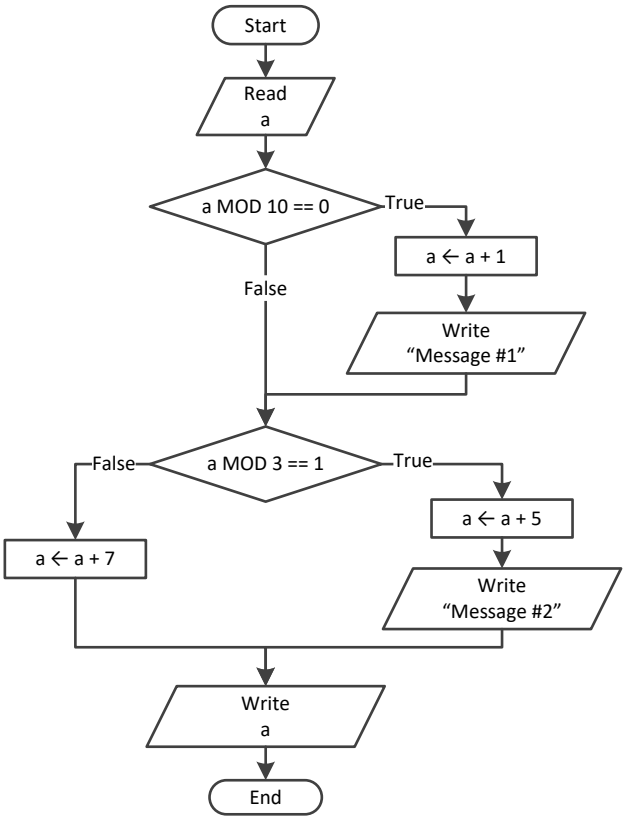
Chapter 21

21.4 Review Exercises

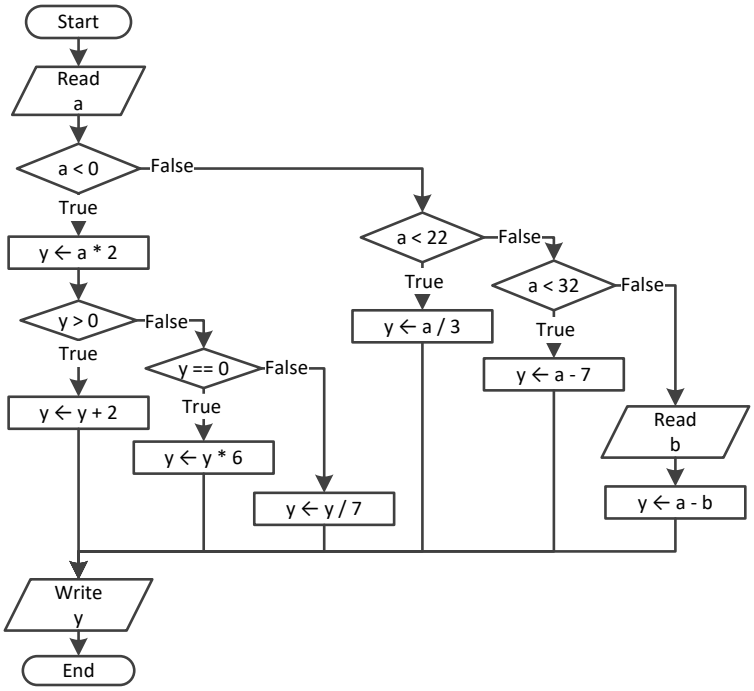
1. Solution



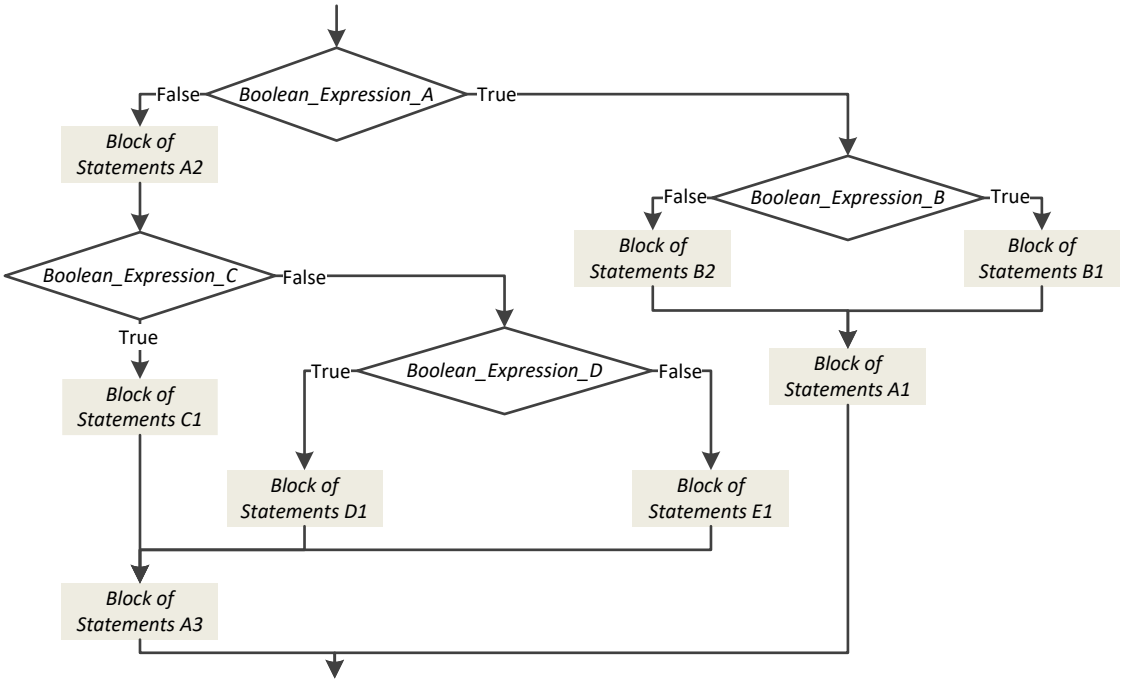
2. Solution



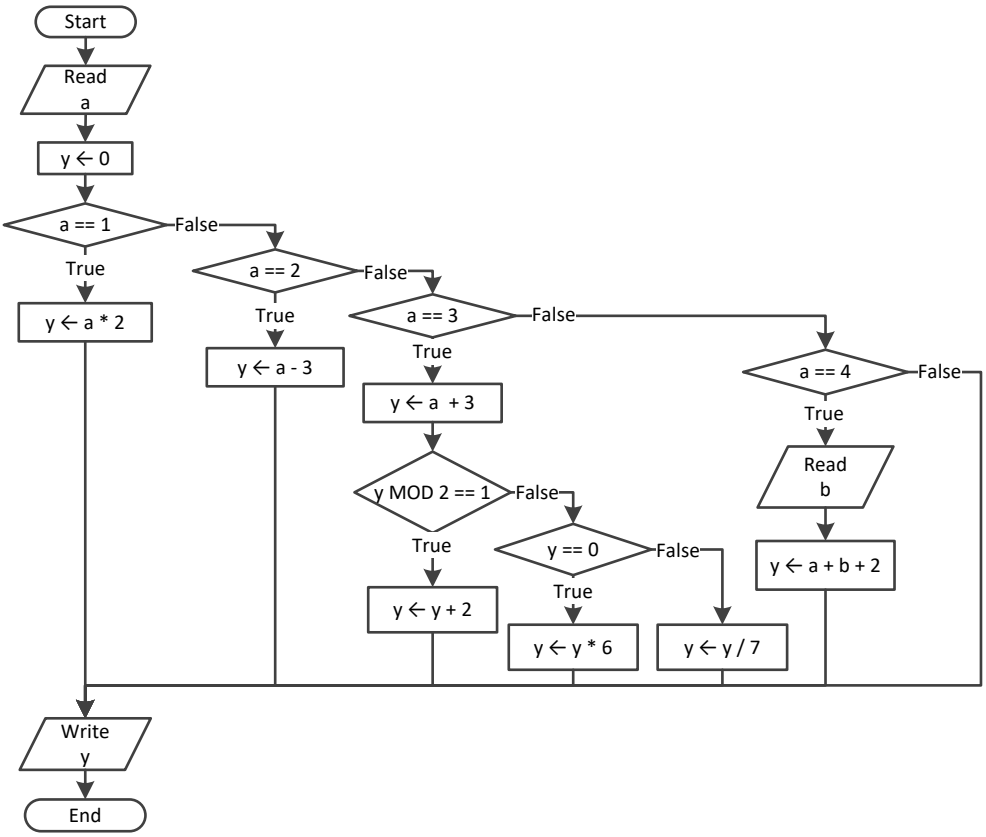
3. Solution



4. Solution



5. Solution



6. Solution

```
double x, y, z;

x = Convert.ToDouble(Console.ReadLine());
y = Convert.ToDouble(Console.ReadLine());

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

7. Solution

```
int x;

x = Convert.ToInt32(Console.ReadLine());

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

8. Solution

```
int a, b, c, d, y;

a = Convert.ToInt32(Console.ReadLine());
b = Convert.ToInt32(Console.ReadLine());

c = a % 2;
d = (int)(b / 5);

if (a >= b)
  y = 1;
else if (d > c && a > 2)
```

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

Console.WriteLine(y);
```

9. Solution

```
int x;

x = Convert.ToInt32(Console.ReadLine());

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

10. Solution

```
double a, b, y;

a = Convert.ToDouble(Console.ReadLine());
b = Convert.ToDouble(Console.ReadLine());

y = a * b;

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

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

11. Solution

```
double a, b, c;

a = Convert.ToDouble(Console.ReadLine());
b = Convert.ToDouble(Console.ReadLine());
c = Convert.ToDouble(Console.ReadLine());

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

Chapter 22

22.9 Review Questions: True/False

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

22.10 Review Questions: Multiple Choice

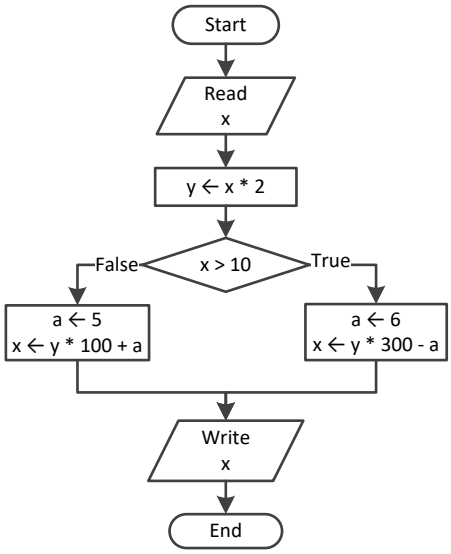
1. a
2. b
3. a
4. c

22.11 Review Exercises

1. Solution

```
int a, x, y;

y = Convert.ToInt32(Console.ReadLine());
x = Convert.ToInt32(Console.ReadLine());

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

2. Solution



3. Solution

```
double a, y;

a = Convert.ToDouble(Console.ReadLine());

if (a >= 10) {
  Console.WriteLine("Error!");
}
else {
  if (a < 1) {
    y = 5 + a;
  }
  else if (a < 5) {
    y = 23 / a;
```

```
  }
  else {
    y = 5 * a;
  }
  Console.WriteLine(y);
}
```

4. Solution

```
int day, month;
string name;

day = Convert.ToInt32(Console.ReadLine());
month = Convert.ToInt32(Console.ReadLine());
name = Console.ReadLine();

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

5. Solution

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

```
double a, b, c, d;

a = Convert.ToDouble(Console.ReadLine());
b = Convert.ToDouble(Console.ReadLine());
c = Convert.ToDouble(Console.ReadLine());

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

6. Solution

```
double a, b, c, d;

a = Convert.ToDouble(Console.ReadLine());
b = Convert.ToDouble(Console.ReadLine());
c = Convert.ToDouble(Console.ReadLine());
```

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

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

7. Solution

```
int a, b, y;

a = Convert.ToInt32(Console.ReadLine());
b = Convert.ToInt32(Console.ReadLine());

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

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

8. Solution

```
double a, b, y;

a = Convert.ToDouble(Console.ReadLine());
b = Convert.ToDouble(Console.ReadLine());

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

9. Solution

```
string os;

Console.Write("What is your tablet's OS? ");
os = Console.ReadLine();

if (os == "iOS") {
  Console.WriteLine("Apple");
}
else if (os == "Android") {
  Console.WriteLine("Google");
}
```

```
else if (os == "Windows") {
  Console.WriteLine("Microsoft");
}
```

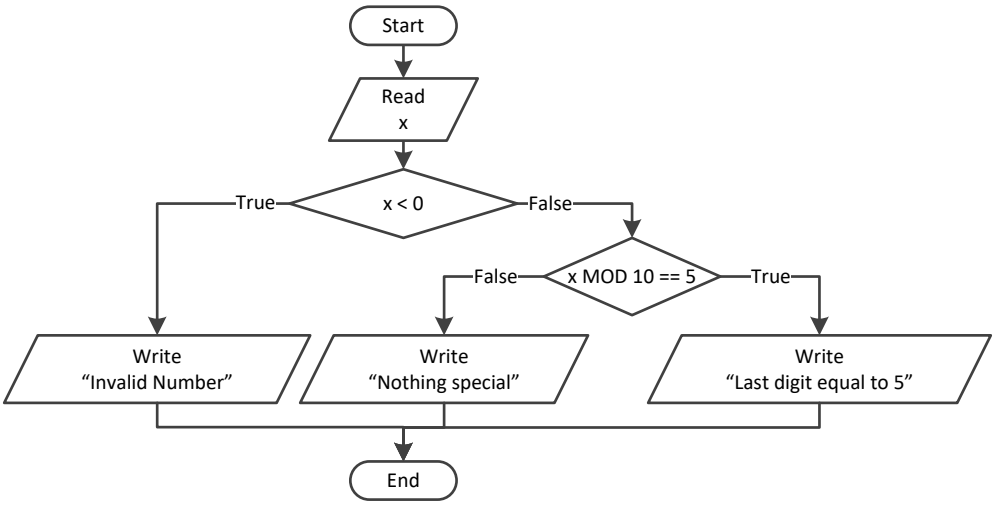
Chapter 23

23.7 Review Exercises

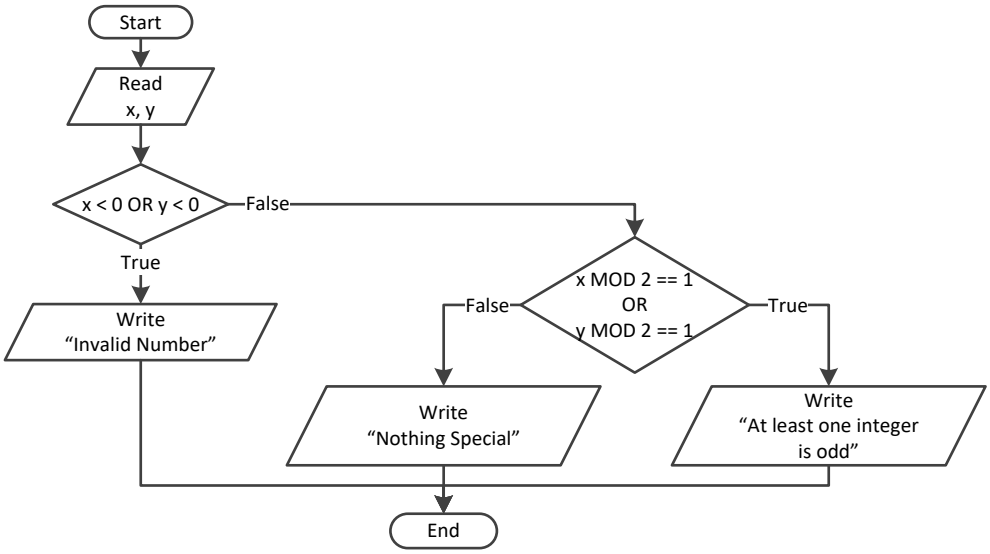
1. Solution

```
double x;

Console.Write("Enter a non-negative number: ");
x = Convert.ToDouble(Console.ReadLine());
if (x < 0) {
  Console.WriteLine("Error! You entered a negative value");
}
else {
  Console.WriteLine("The square root of " + x + " is " + Math.Sqrt(x));
}
```

2. Solution

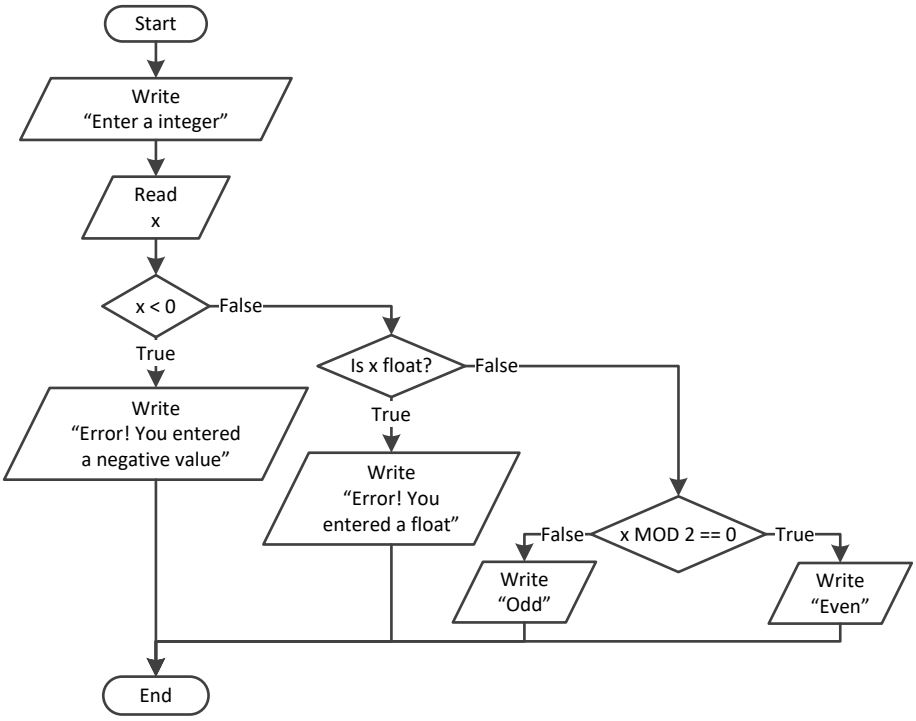


3. Solution



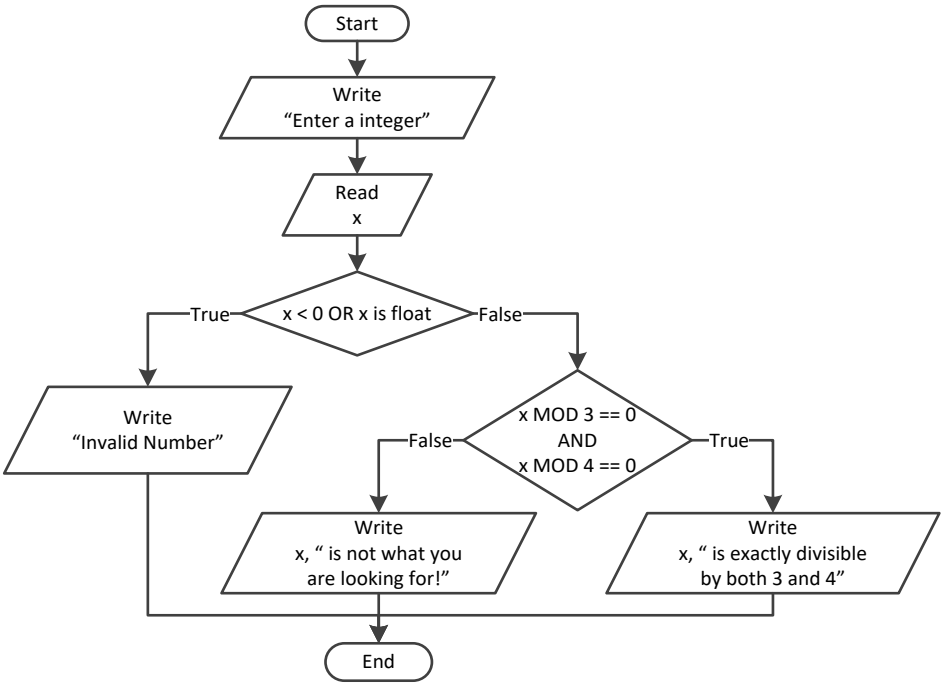

```
int x, y;

x = Convert.ToInt32(Console.ReadLine());
y = Convert.ToInt32(Console.ReadLine());

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

4. Solution



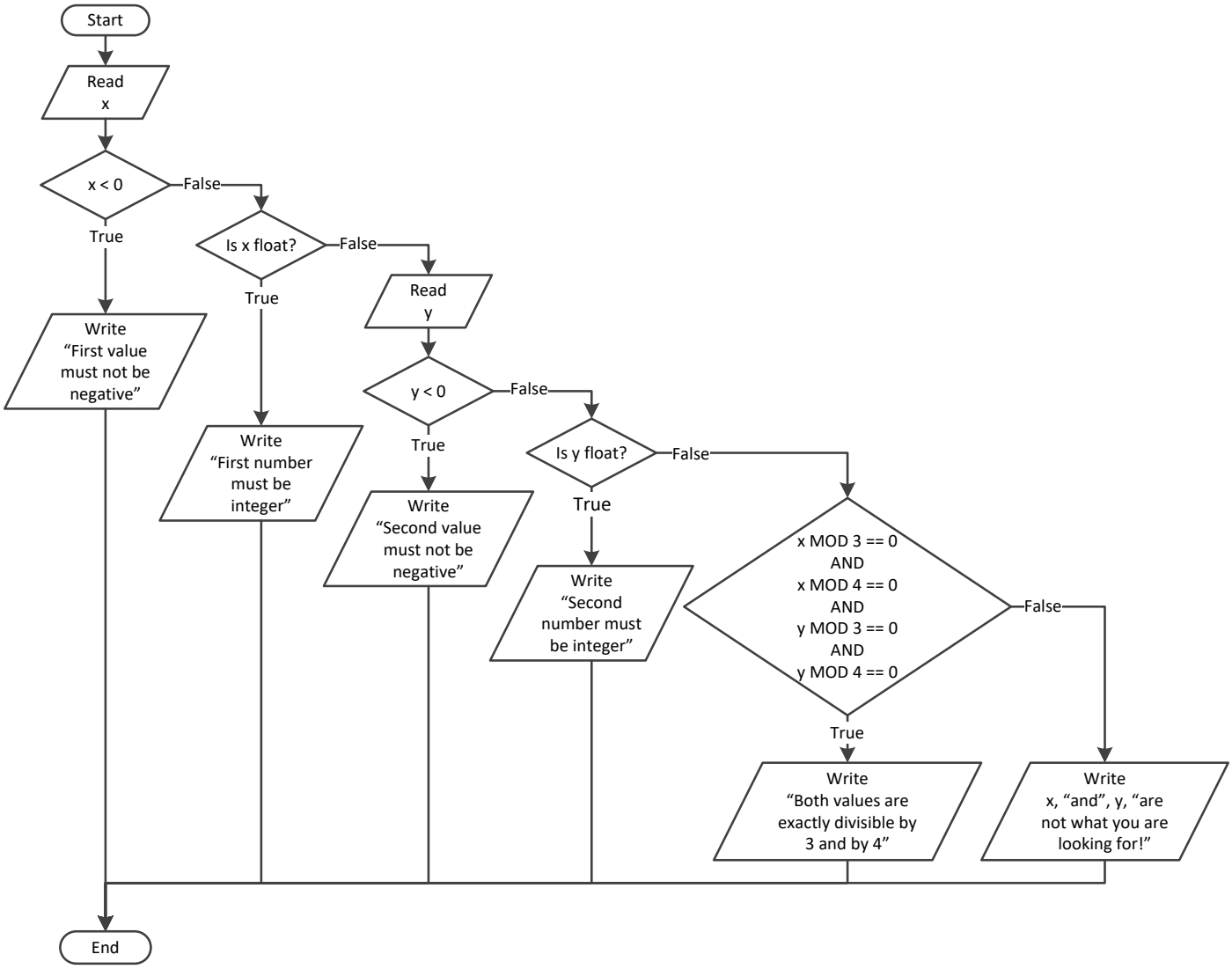

```
double x;

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

5. Solution




```
double x;

Console.Write("Enter an integer: ");
x = Convert.ToDouble(Console.ReadLine());

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

6. Solution




```
double x, y;

x = Convert.ToDouble(Console.ReadLine());

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

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

7. Solution

```
int choice;
double t;

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

Console.Write("Enter a choice: ");
choice = Convert.ToInt32(Console.ReadLine());
Console.Write("Enter a temperature: ");
t = Convert.ToDouble(Console.ReadLine());

if (choice < 1 || choice > 4) {
  Console.WriteLine("Wrong choice");
}
else {
  switch (choice) {
    case 1:
      if (t < 0) {  //Absolute zero in Kelvin
        Console.WriteLine("Wrong temperature");
      }
      else {
        Console.WriteLine(1.8 * t - 459.67);
      }
      break;
    case 2:
      if (t < -459.67) {  //Absolute zero in Fahrenheit
        Console.WriteLine("Wrong temperature");
      }
      else {
        Console.WriteLine((t + 459.57) / 1.8);
      }
      break;
    case 3:
```

```
      if (t < -459.67) {  //Absolute zero in Fahrenheit
        Console.WriteLine("Wrong temperature");
      }
      else {
        Console.WriteLine(5.0 / 9 * (t - 32));
      }
      break;
    case 4:
      if (t < -273.15) {  //Absolute zero in Celcius
        Console.WriteLine("Wrong temperature");
      }
      else {
        Console.WriteLine(9.0 / 5 * t + 32);
      }
      break;
  }
}
```

8. Solution

```
string op, message;
int a, b;

Console.Write("Enter 1st integer: ");
a = Convert.ToInt32(Console.ReadLine());
Console.Write("Enter type of operation: ");
op = Console.ReadLine().ToUpper();
Console.Write("Enter 2nd integer: ");
b = Convert.ToInt32(Console.ReadLine());

message = "The result of " + a + " " + op + " " + b + " equals ";

switch (op) {
  case "+":
    message += a + b;  //Concatenate
    break;
  case "-":
    message += a - b;  //Concatenate
    break;
  case "*":
    message += a * b;  //Concatenate
    break;
  case "/":
    if (b == 0) {
      message = "Infinite";  //Replace
    }
    else {
      message += (double)a / b;  //Concatenate
    }
    break;
  case "DIV":
    if (b == 0) {
```

```
      message = "Infinite";  //Replace
    }
    else {
      message += (int)(a / b);  //Concatenate
    }
    break;
  case "MOD":
    if (b == 0) {
      message = "Infinite";  //Replace
    }
    else {
      message += a % b;  //Concatenate
    }
    break;
  case "POWER":
    message += Math.Pow(a, b);  //Concatenate
    break;
}
Console.WriteLine(message);
```

9. Solution

```
string op, message;
int a, b;

Console.Write("Enter 1st integer: ");
a = Convert.ToInt32(Console.ReadLine());
Console.Write("Enter type of operation: ");
op = Console.ReadLine().ToUpper();
Console.Write("Enter 2nd integer: ");
b = Convert.ToInt32(Console.ReadLine());

message = "The result of " + a + " " + op + " " + b + " equals ";

switch (op) {
  case "+":
    message += a + b;  //Concatenate
    break;
  case "-":
    message += a - b;  //Concatenate
    break;
  case "*":
    message += a * b;  //Concatenate
    break;
  case "/":
    if (b == 0) {
      message = "Infinite";  //Replace
    }
    else {
      message += (double)a / b;  //Concatenate
    }
    break;
```

```
  case "DIV":
    if (b == 0) {
      message = "Infinite";  //Replace
    }
    else {
      message += (int)(a / b);  //Concatenate
    }
    break;
  case "MOD":
    if (b == 0) {
      message = "Infinite";  //Replace
    }
    else {
      message += a % b;  //Concatenate
    }
    break;
  case "POWER":
    message += Math.Pow(a, b);  //Concatenate
    break;
  default:
    message = "Error: Invalid operator";  //Replace
    break;
}
Console.WriteLine(message);
```

10. Solution

```
int a1, a2, a3, maximum, minimum;
string maxName, minName, n1, n2, n3;

Console.Write("Enter the age of the first person: ");
a1 = Convert.ToInt32(Console.ReadLine());
Console.Write("Enter the name of the first person: ");
n1 = Console.ReadLine();
Console.Write("Enter the age of the second person: ");
a2 = Convert.ToInt32(Console.ReadLine());
Console.Write("Enter the name of the second person: ");
n2 = Console.ReadLine();
Console.Write("Enter the age of the third person: ");
a3 = Convert.ToInt32(Console.ReadLine());
Console.Write("Enter the name of the third person: ");
n3 = Console.ReadLine();

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

```
}

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

Console.WriteLine(minName + " " + maxName);
```

11. Solution

```
string artistName;
int score1, score2, score3, score4, score5, minimum, maximum, totalScore;

Console.Write("Enter artist's name: ");
artistName = Console.ReadLine();
Console.Write("Enter score No 1: ");
score1 = Convert.ToInt32(Console.ReadLine());
Console.Write("Enter score No 2: ");
score2 = Convert.ToInt32(Console.ReadLine());
Console.Write("Enter score No 3: ");
score3 = Convert.ToInt32(Console.ReadLine());
Console.Write("Enter score No 4: ");
score4 = Convert.ToInt32(Console.ReadLine());
Console.Write("Enter score No 5: ");
score5 = Convert.ToInt32(Console.ReadLine());

minimum = score1;
if (score2 < minimum) {
  minimum = score2;
}
if (score3 < minimum) {
  minimum = score3;
}
if (score4 < minimum) {
  minimum = score4;
}
if (score5 < minimum) {
  minimum = score5;
}

maximum = score1;
if (score2 > maximum) {
  minimum = score2;
}
if (score3 > maximum) {
```

```
  minimum = score3;
}
if (score4 > maximum) {
  minimum = score4;
}
if (score5 > maximum) {
  minimum = score5    ;
}

totalScore = score1 + score2 + score3 + score4 + score5 - minimum - maximum;
Console.WriteLine(artistName + " received " + totalScore + " points");
```

12. Solution

```
int age1, age2, age3, maximum, middle, minimum;

Console.Write("Enter age for person No1:");
age1 = Convert.ToInt32(Console.ReadLine());
Console.Write("Enter age for person No2:");
age2 = Convert.ToInt32(Console.ReadLine());
Console.Write("Enter age for person No3:");
age3 = Convert.ToInt32(Console.ReadLine());

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

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

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

13. Solution

```
int a1, a2, a3, maximum, minimum, middle;
string maxName, minName, n1, n2, n3;

Console.Write("Enter the age of the first person: ");
a1 = Convert.ToInt32(Console.ReadLine());
Console.Write("Enter the name of the first person: ");
n1 = Console.ReadLine();
Console.Write("Enter the age of the second person: ");
a2 = Convert.ToInt32(Console.ReadLine());
```

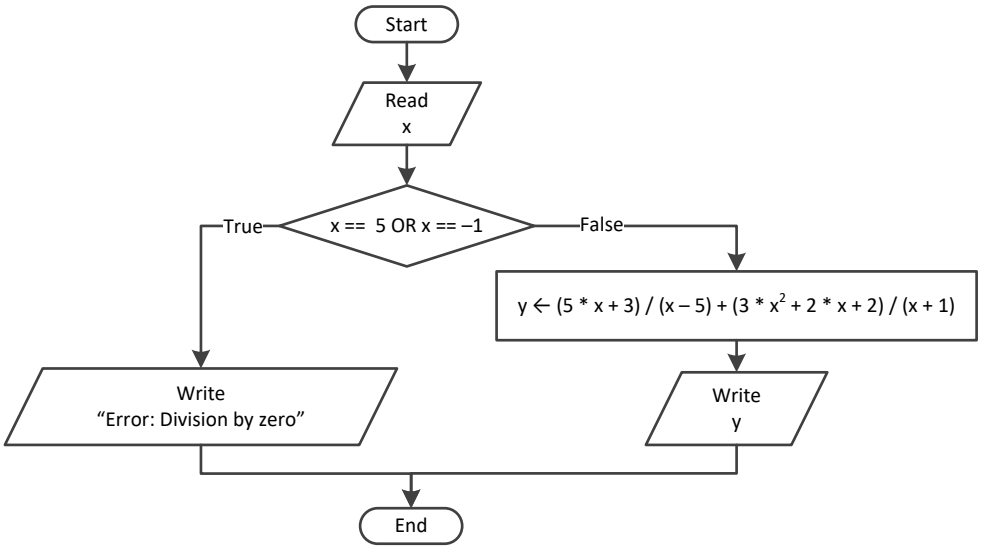
```
Console.Write("Enter the name of the second person: ");
n2 = Console.ReadLine();
Console.Write("Enter the age of the third person: ");
a3 = Convert.ToInt32(Console.ReadLine());
Console.Write("Enter the name of the third person: ");
n3 = Console.ReadLine();

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

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

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

if (Math.Abs(maximum - middle) < Math.Abs(minimum - middle)) {
  Console.WriteLine(maxName);
}
else {
  Console.WriteLine(minName);
}
```

14. Solution

```
string title1, title2, title3, minName;
double price1, price2, price3, minimum, amount;

price1 = Convert.ToDouble(Console.ReadLine());
title1 = Console.ReadLine();
price2 = Convert.ToDouble(Console.ReadLine());
title2 = Console.ReadLine();
price3 = Convert.ToDouble(Console.ReadLine());
title3 = Console.ReadLine();

minimum = price1;
minName = title1;
if (price2 < minimum) {
```

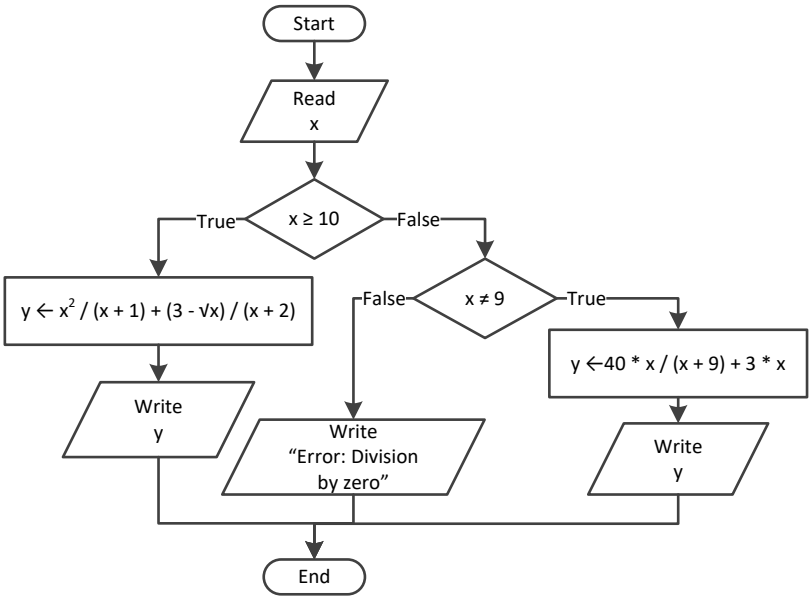
```
  minimum = price2;
  minName = title2;
}
if (price3 < minimum) {
  minimum = price3;
  minName = title3;
}

amount = price1 + price2 + price3 - minimum;
Console.WriteLine("You need to pay: $" + amount);
Console.WriteLine("Title provided for free: " + minName);
Console.WriteLine("You saved: $" + minimum);
```

15. Solution




```
double x, y;

x = Convert.ToDouble(Console.ReadLine());

if (x == 5 || x == -1) {
  Console.WriteLine("Error: Division by zero");
}
else {
  y = (5 * x + 3) / (x - 5) + (3 * Math.Pow(x, 2) + 2 * x + 2) / (x + 1);
  Console.WriteLine(y);
}
```

16. Solution




```
double x, y;

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

17. Solution

```
double x, y;

x = Convert.ToDouble(Console.ReadLine());

if (x < 0 ) {
  y = 40 * x / (x - 5) + 3;
  Console.WriteLine(y);
}
else if (x == 0 || x == 3) {
  Console.WriteLine("Error: Division by zero!");
}
else {
  y = (7 + x) / (x - 3) + (3 - x) / x;
  Console.WriteLine(y);
}
```

18. Solution

```
double x, y;

x = Convert.ToDouble(Console.ReadLine());
if (x <= -15 || x > 25) {
  y = x - 1;
  Console.WriteLine(y);
}
else if (x <= -10) {
  y = x / Math.Sqrt(x + 30) + Math.Pow(8 + x, 2) / (x + 1);
  Console.WriteLine(y);
}
else if (x <= 0) {
  y = Math.Abs(40 * x) / (x - 8);
  Console.WriteLine(y);
}
else {
  if (x == 9) {
    Console.WriteLine("Error: Division by zero");
  }
  else if (x < 9) {
    Console.WriteLine("Error: Invalid square root");
  }
  else {
    y = 3 * x / Math.Sqrt(x - 9);
    Console.WriteLine(y);
  }
}
```

19. Solution

```
int digit1, digit2, digit3, r, total;
double x;

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

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

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

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

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

20. Solution

```
int d, m, y;

Console.Write("Enter day 1 - 31: ");
d = Convert.ToInt32(Console.ReadLine());
Console.Write("Enter month 1 - 12: ");
m = Convert.ToInt32(Console.ReadLine());
Console.Write("Enter year: ");
y = Convert.ToInt32(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);
}
```

21. Solution

First approach

```
string word, word1, word2;

word = Console.ReadLine();

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

word2 = word.Substring(0, 1).ToLower() +
        word.Substring(1, 1).ToUpper() +
        word.Substring(2, 1).ToLower() +
        word.Substring(3, 1).ToUpper() +
```

```
        word.Substring(4, 1).ToLower() +
        word.Substring(5, 1).ToUpper();

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

Second approach

```
string word, word1, word2;

word = Console.ReadLine();

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

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

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

22. Solution

```
int q;
double discount, payment;

Console.Write("Enter quantity: ");
q = Convert.ToInt32(Console.ReadLine());

if (q < 3) {
  discount = 0;
}
else if (q < 6) {
  discount = 10;
}
else if (q < 10) {
  discount = 15;
}
else if (q < 14) {
```

```
  discount = 20;
}
else if (q < 20) {
  discount = 27;
}
else {
  discount = 30;
}

payment = q * 10 - q * 10 * discount / 100.0;

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

23. Solution

```
const double VAT = 0.19;

double amount, discount, payment;

Console.Write("Enter a before-tax amount: : ");
amount = Convert.ToDouble(Console.ReadLine());

if (amount < 0) {
  Console.WriteLine("Error! You entered a negative value");
}
else {
  if (amount < 50) {
    discount = 0;
  }
  else if (amount < 100) {
    discount = 1;
  }
  else if (amount < 250) {
    discount = 2;
  }
  else {
    discount = 3;
  }

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

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

24. Solution

```
int a, h, w;
double bmi;

Console.Write("Enter age: ");
```

```
a = Convert.ToInt32(Console.ReadLine());
if (a < 18) {
  Console.WriteLine("Invalid age");
}
else {
  Console.Write("Enter weight in pounds: ");
  w = Convert.ToInt32(Console.ReadLine());
  Console.Write("Enter height in inches: ");
  h = Convert.ToInt32(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");
  }
}
```

25. Solution

```
const double TAX_RATE = 0.10;

int water;
double total;

Console.Write("Enter water consumption (in cubic feet): ");
water = Convert.ToInt32(Console.ReadLine());

if (water < 0) {
  Console.WriteLine("Error! You entered a negative value");
}
else {
  if (water <= 10) {
    total = water * 3;
  }
```

```
  else if (water <= 20) {
    total = 10 * 3 + (water - 10) * 5;
  }
  else if (water <= 35) {
    total = 10 * 3 + 10 * 5 + (water - 20) * 7;
  }
  else {
    total = 10 * 3 + 10 * 5 + 15 * 7 + (water - 35) * 9;
  }

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

26. Solution

```
int children;
double income, tax;

Console.Write("Enter taxable income: ");
income = Convert.ToDouble(Console.ReadLine());
Console.Write("Enter number of children: ");
children = Convert.ToInt32(Console.ReadLine());

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

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

27. Solution

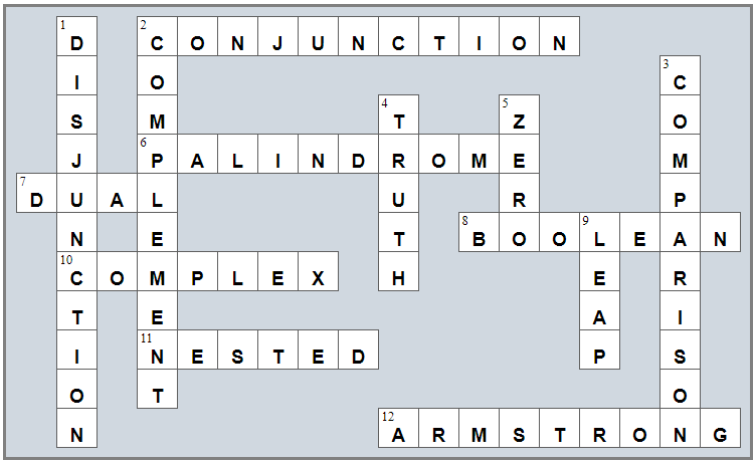
```
double wind;

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

if (wind < 0) {
  Console.WriteLine("Error! You entered a negative value");
}
else {
```

```
  if (wind < 1) {
    Console.WriteLine("Beaufort: 0\nCalm");
  }
  else if (wind < 4) {
    Console.WriteLine("Beaufort: 1\nLight air");
  }
  else if (wind < 8) {
    Console.WriteLine("Beaufort: 2\nLight breeze");
  }
  else if (wind < 13) {
    Console.WriteLine("Beaufort: 3\nGentle breeze");
  }
  else if (wind < 18) {
    Console.WriteLine("Beaufort: 4\nModerate breeze");
  }
  else if (wind < 25) {
    Console.WriteLine("Beaufort: 5\nFresh breeze");
  }
  else if (wind < 31) {
    Console.WriteLine("Beaufort: 6\nStrong breeze");
  }
  else if (wind < 39) {
    Console.WriteLine("Beaufort: 7\nModerate gale");
  }
  else if (wind < 47) {
    Console.WriteLine("Beaufort: 8\nGale");
  }
  else if (wind < 55) {
    Console.WriteLine("Beaufort: 9\nStrong gale");
  }
  else if (wind < 64) {
    Console.WriteLine("Beaufort: 10\nStorm");
  }
  else if (wind < 74) {
    Console.WriteLine("Beaufort: 11\nViolent storm");
  }
  else {
    Console.WriteLine("Beaufort: 12\nHurricane force");
  }

  if (wind < 13) {
    Console.WriteLine("It's Fishing Day!!!");
  }
}
```

Review in “Decision Control Structures”

Review Crossword Puzzle

1.



Chapter 24

24.3 Review Questions: True/False

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

Chapter 25

25.4 Review Questions: True/False

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

25.5 Review Questions: Multiple Choice

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

25.6 Review Exercises

1. Solution

```
int i;

i = 3;
do {
  Console.WriteLine(i);
  i--;
} while (i > 0);   //Alternatively you can use the logical operator !=
Console.WriteLine("The end");
```

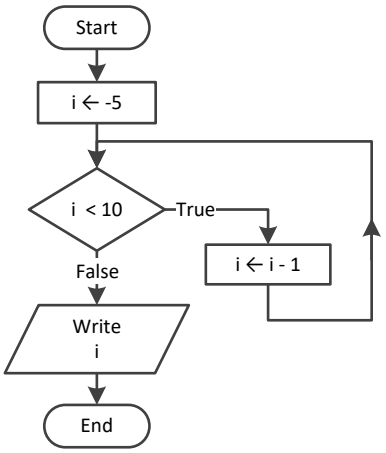
2. Solution

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

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

It performs 4 iterations

3. Solution



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

It performs an infinite number of iterations

4. Solution

Step	Statement	a	b	c	d
1	`a = 2`	**2**	?	?	?
2	`while (a <= 10)`	true			
3	`b = a + 1`	2	**3**	?	?
4	`c = b * 2`	2	3	**6**	?

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

5. Solution

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

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

6. Solution

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

7. Solution

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

8. Solution

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

9. 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.WriteLine(x + ", " + y)`	It displays: 4, 4	
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.WriteLine(x + ", " + y)`	It displays: 20, 16	
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.WriteLine(x + ", " + y)`	It displays: 276, 256	
17	`while (y < 65535)`	true	
18	`y = Math.Pow (y, 2)`	276	**65536**
19	`if (x < 256)`	false	
20	`Console.WriteLine(x + ", " + y)`	It displays: 276, 65536	
21	`while (y < 65535)`	false	

10. 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	`while (c < 11)`	true				
25	`x = a + b`	5	5	8	5	**10**
26	`if (x % 2 != 0)`	false				
27	`else if (d % 2 == 0)`	false				
28	`c = c + 3`	5	5	**11**	5	10
29	`a = b`	**5**	5	11	5	10
30	`b = d`	5	**5**	11	5	10
31	`while (c < 11)`	false				

11. Solution

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

12. Solution

i. 4
ii. −2
iii. 2
iv. 10

13. Solution

```
double a, total;
int i, n;

n = Convert.ToInt32(Console.ReadLine());
total = 0

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

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

14. Solution

```
int a, i, n, p;
int count = 0;

n = Convert.ToInt32(Console.ReadLine());
p = 1;

i = 1;
while (i <= n) {
  a = Convert.ToInt32(Console.ReadLine());
  if (a % 2 == 0) {
    p = p * a;
    count++;
  }
  i++;
}

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

15. Solution

```
int a, i, total;

total = 0;

i = 1;
while (i <= 100) {
  a = Convert.ToInt32(Console.ReadLine());
  if (a % 10 == 0) {
    total = total + a;
  }
  i++;
}
Console.WriteLine(total);
```

16. Solution

```
int a, i, total;

total = 0;

i = 1;
```

```
while (i <= 20) {
  a = Convert.ToInt32(Console.ReadLine());
  if (a >= 100 && a <= 999) {
    total = total + a;
  }
  i++;
}
Console.WriteLine(total);
```

17. Solution

```
double a, p;

p = 1;

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

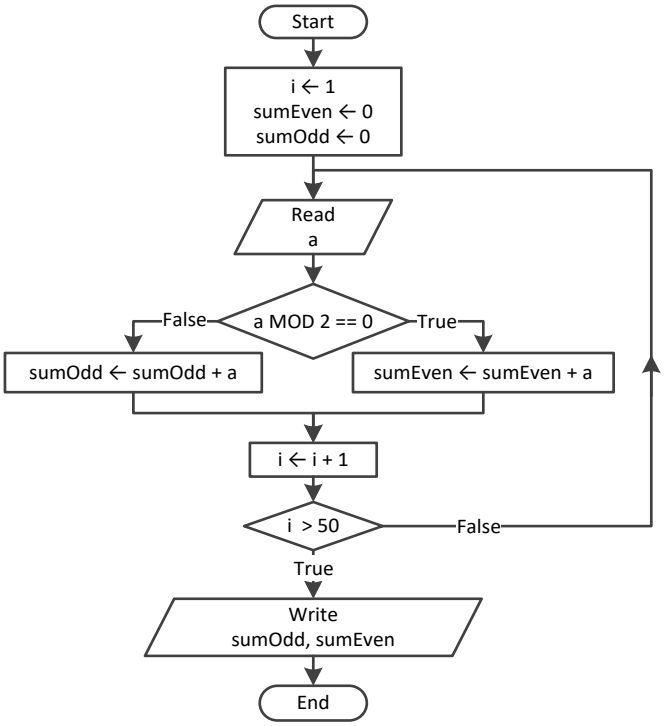
Step	Statement	a	p
1	p = 1	?	**1.0**
2	a = Convert.ToDouble(Console.ReadLine())	**3.0**	1.0
3	while (a != 0)	true	
4	p = p * a	3.0	**3.0**
5	a = Convert.ToDouble(Console.ReadLine())	**2.0**	3.0
6	while (a != 0)	true	
7	p = p * a	2.0	**6.0**
8	a = Convert.ToDouble(Console.ReadLine())	**9.0**	6.0
9	while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

18. Solution

```
int years;
double population;

population = 30000;

years = 0;
while (population <= 100000) {
  population += population * 0.03;
  years++;
```

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

19. Solution




```
int a, i, sumEven, sumOdd;

i = 1;
sumEven = 0;
sumOdd = 0;
do {
  a = Convert.ToInt32(Console.ReadLine());
  if (a % 2 == 0) {
    sumEven += a;
  }
  else {
    sumOdd += a;
  }
  i++;
} while (i <= 50);
Console.WriteLine(sumEven + " " + sumOdd);
```

20. Solution

```
int a, i, n, p;

n = Convert.ToInt32(Console.ReadLine());
i = 1;
p = 1;
do {
  a = Convert.ToInt32(Console.ReadLine());
```

```
  if (a < 0) {
    p *= a;
  }
  i++;
} while (i <= n);
Console.WriteLine(Math.Abs(p));
```

21. Solution

```
int a, i, p;

i = 1;
p = 1;
do {
  Console.Write("Enter an integer: ");
  a = Convert.ToInt32(Console.ReadLine());
  if (a >= 500 && a <= 599) {
    p *= a;
  }
  i++;
} while (i <= 5);
Console.WriteLine(p);
```

22. Solution

```
double population;
int years;

population = 50000;

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

Chapter 26

26.3 Review Questions: True/False

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

26.4 Review Questions: Multiple Choice

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

7. a
8. b
9. c
10. b
11. d
12. d
13. c
14. c

26.5 Review Exercises

1. Solution

Step	Statement	a	b	j
1	`a = 0`	**0**	?	?
2	`b = 0`	0	**0**	?
3	`j = 0`	0	0	**0**
4	`j <= 8`		true	
5	`if (j < 5)`		true	
6	`b++`	0	**1**	0
7	`j += 2`	0	1	**2**
8	`j <= 8`		true	
9	`if (j < 5)`		true	
10	`b++`	0	**2**	2
11	`j += 2`	0	2	**4**
12	`j <= 8`		true	
13	`if (j < 5)`		true	
14	`b++`	0	**3**	4
15	`j += 2`	0	3	**6**
16	`j <= 8`		true	
17	`if (j < 5)`		false	
18	`a += j - 1`	**5**	3	6
19	`j += 2`	5	3	**8**
20	`j <= 8`		true	
21	`if (j < 5)`		false	
22	`a += j - 1`	**12**	3	8
23	`j += 2`	12	3	**10**
24	`j <= 8`		false	
25	`Console.WriteLine(a + ", " + b)`	It displays: 12, 3		

2. Solution

For input value of 10

Step	Statement	a	b	j
1	`a = Convert.ToInt32(Console.ReadLine())`	**10**	?	?

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

For input value of 21

Step	Statement	a	b	j
1	`a = Convert.ToInt32(Console.ReadLine())`	**21**	?	?
2	`b = a`	21	**21**	?
3	`j = a - 5`	21	21	**16**
4	`j <= a`		true	
5	`if (j % 2 != 0)`		false	
6	`b = a + j + 5`	21	**5**	16
7	`j += 2`	21	5	**18**
8	`j <= a`		true	
9	`if (j % 2 != 0)`		false	
10	`b = a + j + 5`	21	**3**	18
11	`j += 2`	21	3	**20**
12	`j <= a`		true	
13	`if (j % 2 != 0)`		false	
14	`b = a + j + 5`	21	**1**	20
15	`j += 2`	21	1	**22**
16	`j <= a`		false	
17	`Console.WriteLine(b)`	It displays: 1		

3. Solution

For input value of 12

Step	Statement	a	x	y	j
1	`a = Convert.ToInt32(Console.ReadLine())`	**12**	?	?	?
2	`j = 2`	12	?	?	**2**
3	`j <= a - 1`	true			
4	`x = j * 3 + 3`	12	**9**	?	2
5	`y = j * 2 + 10`	12	9	**14**	2
6	`if (y - x > 0 || x > 30)`	true			
7	`y *= 2`	12	9	**28**	2
8	`x += 4`	12	**13**	28	2
9	`Console.WriteLine(x + ", " + y)`	It displays: 13, 28			
10	`j += 3`	12	13	28	**5**
11	`j <= a - 1`	true			
12	`x = j * 3 + 3`	12	**18**	28	5
13	`y = j * 2 + 10`	12	18	**20**	5
14	`if (y - x > 0 || x > 30)`	true			
15	`y *= 2`	12	18	**40**	5
16	`x += 4`	12	**22**	40	5
17	`Console.WriteLine(x + ", " + y)`	It displays: 22, 40			
18	`j += 3`	12	22	40	**8**
19	`j <= a - 1`	true			
20	`x = j * 3 + 3`	12	**27**	40	8
21	`y = j * 2 + 10`	12	27	**26**	8
22	`if (y - x > 0 || x > 30)`	false			
23	`x += 4`	12	**31**	26	8
24	`Console.WriteLine(x + ", " + y)`	It displays: 31, 26			
25	`j += 3`	12	31	26	**11**
26	`j <= a - 1`	true			
27	`x = j * 3 + 3`	12	**36**	26	11
28	`y = j * 2 + 10`	12	36	**32**	11
29	`if (y - x > 0 || x > 30)`	true			
30	`y *= 2`	12	36	**64**	11
31	`x += 4`	12	**40**	64	11
32	`Console.WriteLine(x + ", " + y)`	It displays: 40, 64			
33	`j += 3`	12	40	64	**14**
34	`j <= a - 1`	false			

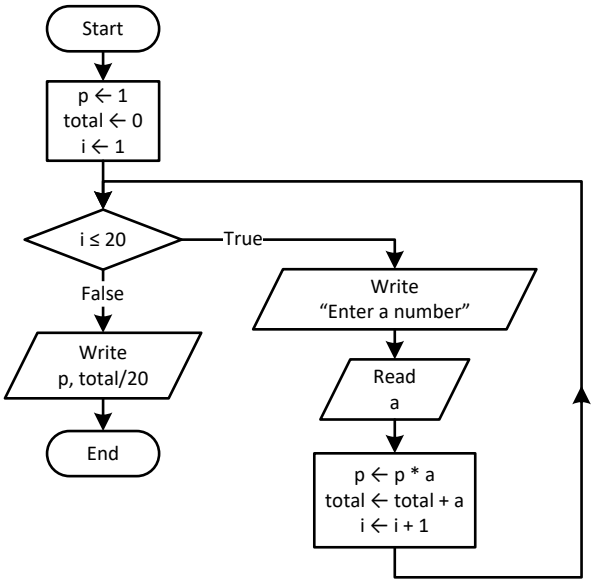
4. Solution

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

5. Solution

It displays: sueZ

6. Solution




```
double a, p, total;
int i;

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

7. Solution

```
double i;

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

8. Solution

```
int deg, i;

Console.Write("Enter degrees: ");
deg = Convert.ToInt32(Console.ReadLine());
for (i = 0 ; i <= deg; i++) {
  Console.WriteLine(Math.Cos(i * Math.PI / 180));
}
```

9. Solution

```
int i, s;

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

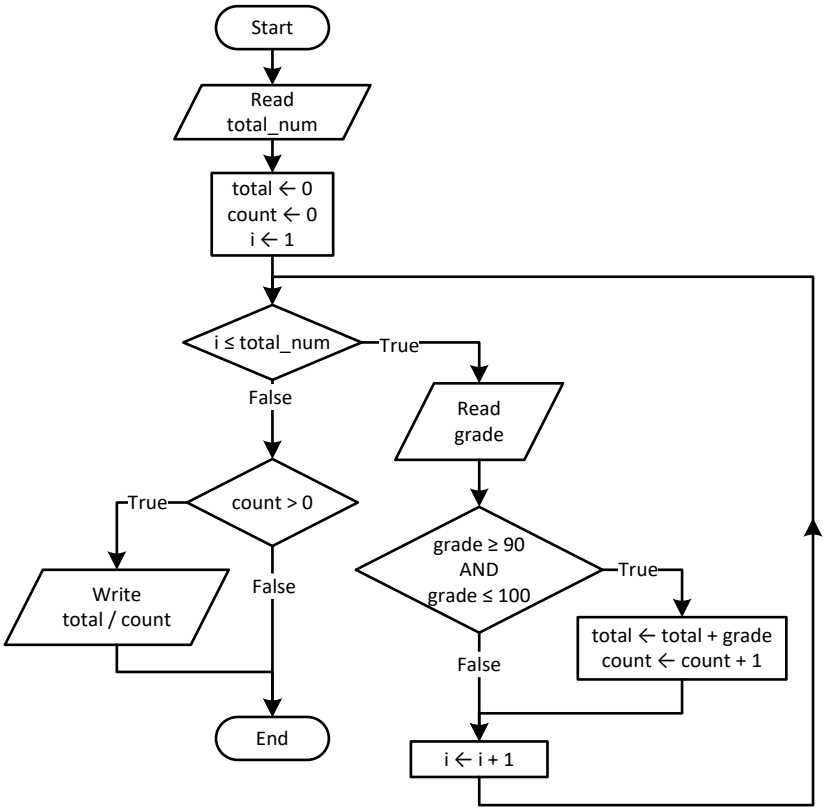
10. Solution

```
int i, n;
double p;

n = Convert.ToInt32(Console.ReadLine());
p = 1;
for (i = 2; i <= 2 * n; i += 2) {
  p *= Math.Pow(i, i - 1);
}
Console.WriteLine(p);
```

11. Solution

```
int i, offset, s;

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

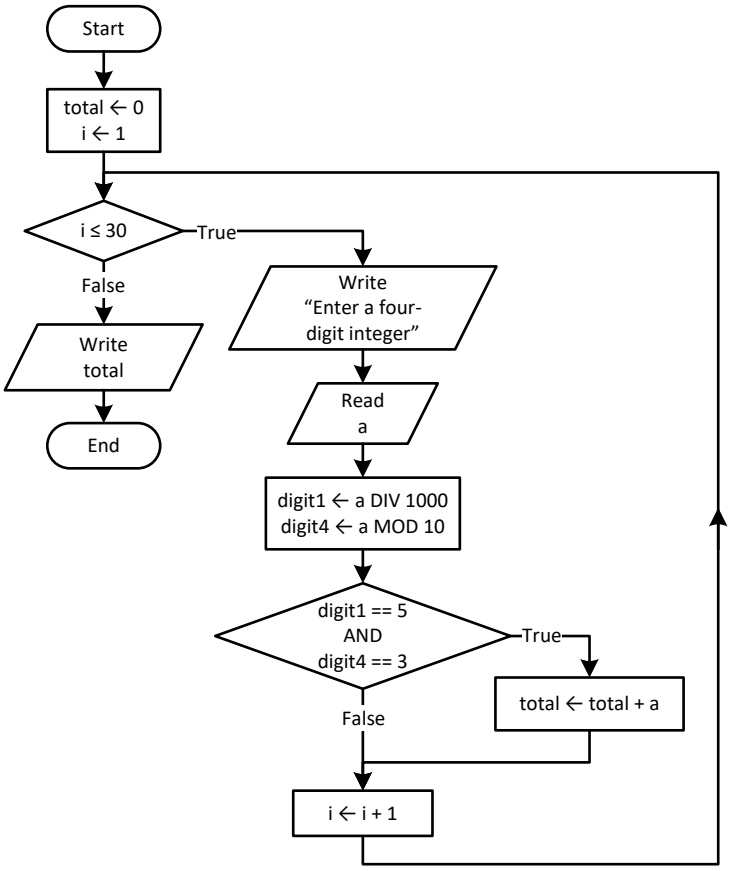
12. Solution




```
int count, grade, i, totalNum, total;

totalNum = Convert.ToInt32(Console.ReadLine());
total = 0;
count = 0;
for (i = 1; i <= totalNum; i++) {
  grade = Convert.ToInt32(Console.ReadLine());
  if (grade >= 90 && grade <= 100) {
    total += grade;
    count++;
  }
}
if (count > 0) {
  Console.WriteLine(total / (double)count);
}
```

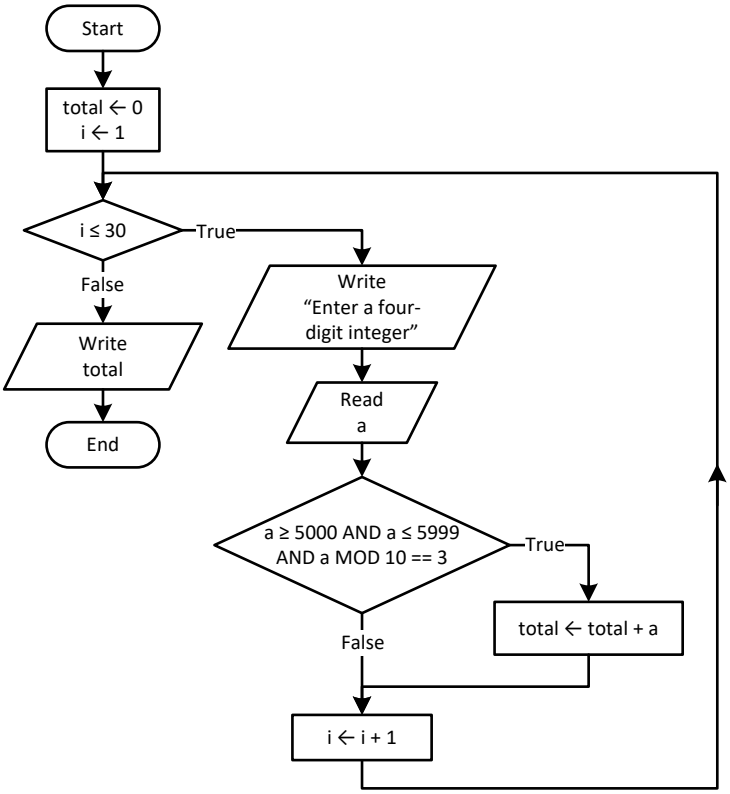
13. Solution

First approach



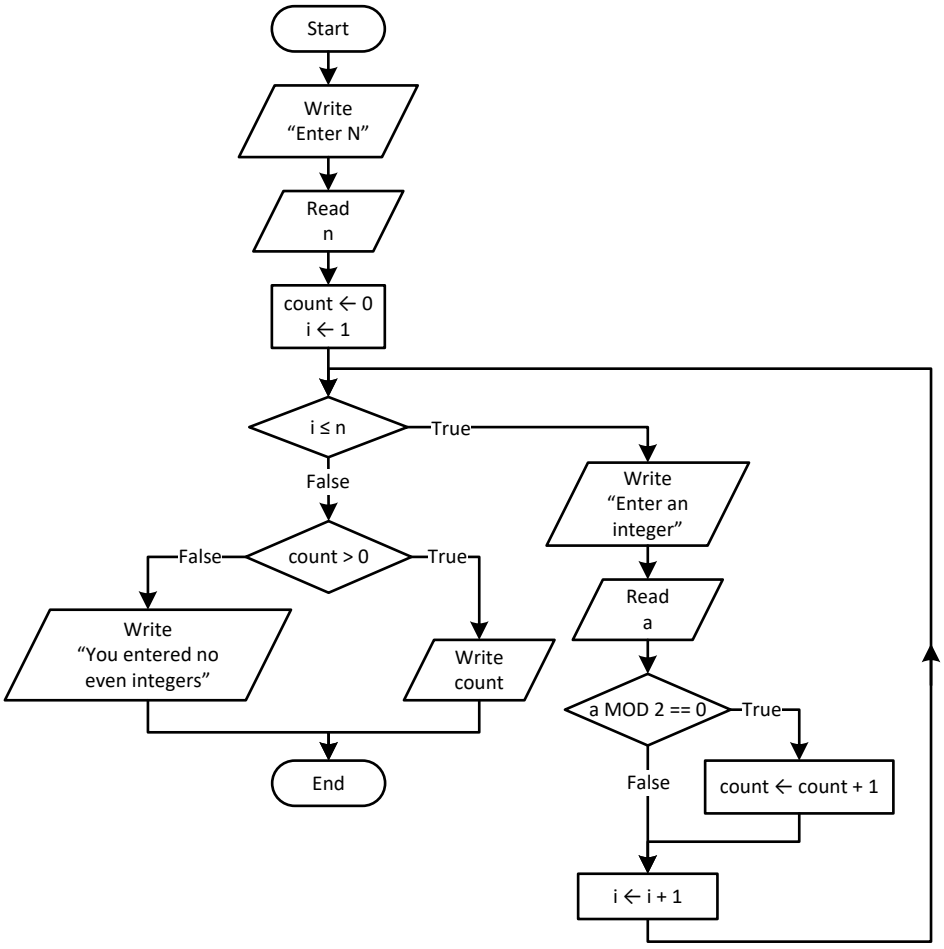

```
int a, digit1, digit4, i, total;

total = 0;
for (i = 1; i <= 30; i++) {
  Console.Write("Enter a four-digit integer: ");
  a = Convert.ToInt32(Console.ReadLine());
  digit1 = (int)(a / 1000);
  digit4 = a % 10;
  if (digit1 == 5 && digit4 == 3) {
    total += a;
  }
}
Console.WriteLine(total);
```

Second approach



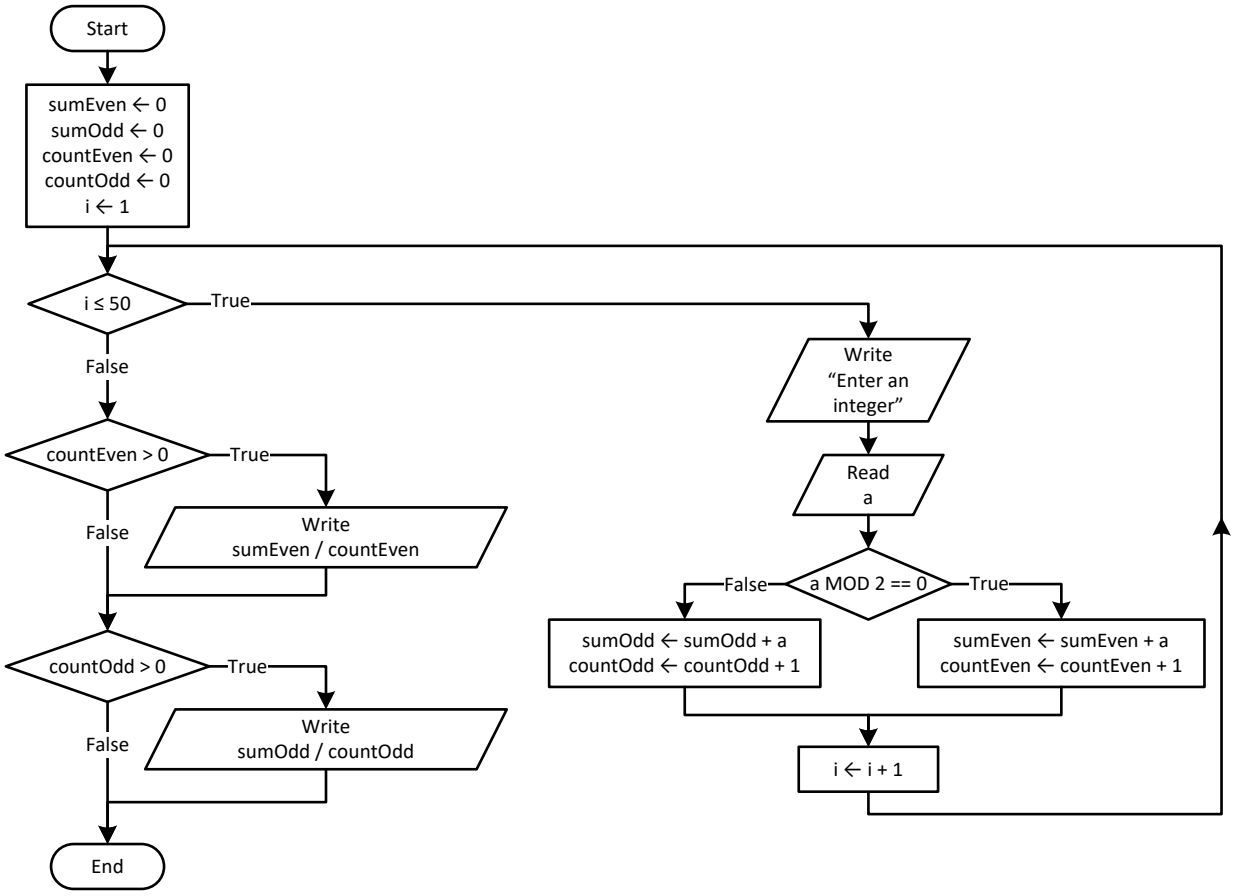

```
int a, i, total;

total = 0;
for (i = 1; i <= 30; i++) {
  Console.Write("Enter a four-digit integer: ");
  a = Convert.ToInt32(Console.ReadLine());
  if (a >= 5000 & a <= 5999 and a % 10 == 3) {
    total += a;
  }
}
Console.WriteLine(total);
```

14. Solution




```
int a, count, i, n;

Console.Write("Enter N: ");
n = Convert.ToInt32(Console.ReadLine());
count = 0;
for (i = 1; i <= n; i++) {
  Console.Write("Enter an integer: ");
  a = Convert.ToInt32(Console.ReadLine());
  if (a % 2 == 0) {
    count++;
  }
}
if (count > 0) {
  Console.WriteLine(count);
}
else {
  Console.WriteLine("You entered no even integers");
}
```

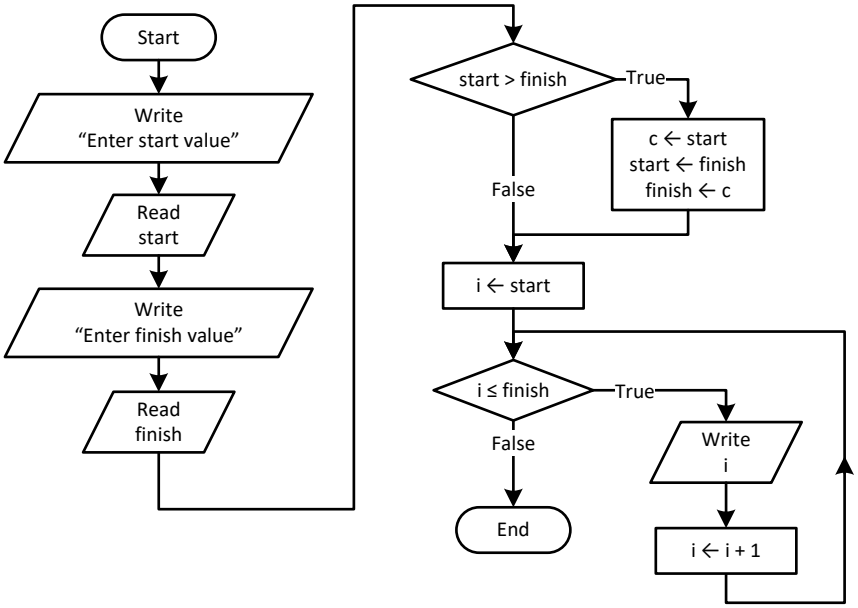
15. Solution




```
int a, countEven, countOdd, i, sumEven, sumOdd;

sumEven = 0;
sumOdd = 0;
countEven = 0;
countOdd = 0;
for (i = 1; i <= 50; i++) {
  Console.Write("Enter an integer: ");
  a = Convert.ToInt32(Console.ReadLine());
  if (a % 2 == 0) {
    sumEven += a;
    countEven++;
  }
  else {
    sumOdd += a;
    countOdd++;
  }
}
if (countEven > 0) {
  Console.WriteLine(sumEven / (double)countEven);
}
if (countOdd > 0) {
  Console.WriteLine(sumOdd / (double)countOdd);
```

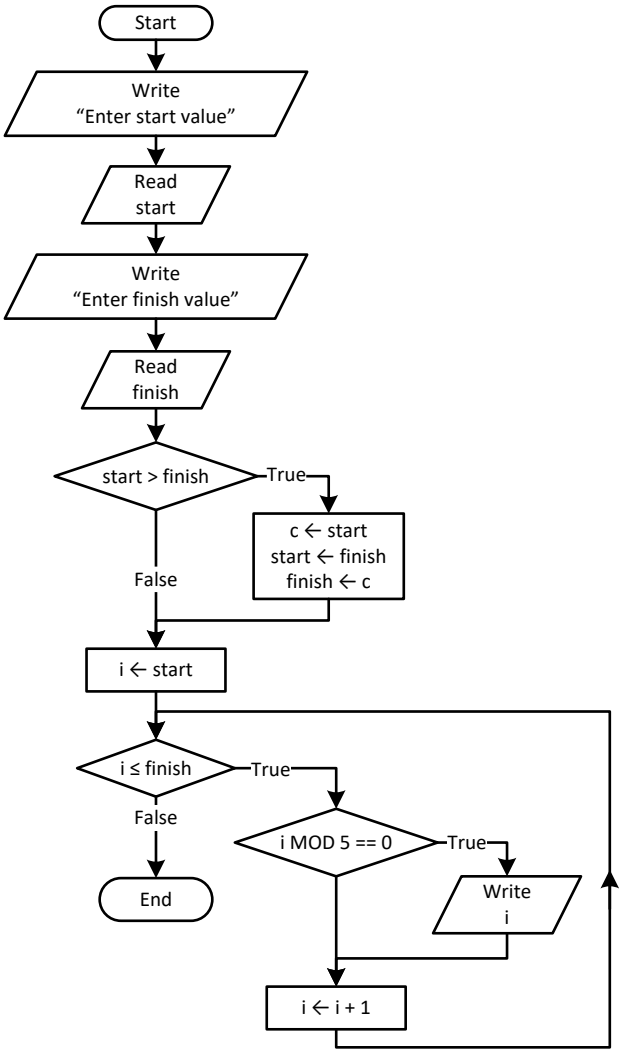
```
}
```

16. Solution




```
int c, finish, i, start;

Console.Write("Enter start value: ");
start = Convert.ToInt32(Console.ReadLine());
Console.Write("Enter finish value: ");
finish = Convert.ToInt32(Console.ReadLine());

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

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

17. Solution




```
int c, finish, i, start;

Console.Write("Enter start value: ");
start = Convert.ToInt32(Console.ReadLine());
Console.Write("Enter finish value: ");
finish = Convert.ToInt32(Console.ReadLine());

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

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

18. Solution

First approach

```
int exp, i;
double p, b;

Console.Write("Enter a value for base: ");
b = Convert.ToDouble(Console.ReadLine());
Console.Write("Enter an integer for exponent: ");
exp = Convert.ToInt32(Console.ReadLine());

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

Second approach

```
int exp, i;
double p, b;

Console.Write("Enter a value for base: ");
b = Convert.ToDouble(Console.ReadLine());
Console.Write("Enter an integer for exponent: ");
exp = Convert.ToInt32(Console.ReadLine());

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

19. Solution

```
int count, i, words;
string msg, character;

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

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

```
  if (character == " ") {
    count++;
  }
}
words = count + 1;

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

20. Solution

```
int characters, count, i, words;
string msg, character;

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

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

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

21. Solution

```
string message;
char character;
string consonants = "BCDFGHJKLMNPQRSTVWZYX";
int i, count;

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

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

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

22. Solution

```
string message;
char character;
```

```
string vowels = "AEIOU";
string consonants = "BCDFGHJKLMNPQRSTVWZYX";
string digits = "0123456789";
int i, countv, countc, countd;

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

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

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

Chapter 27

27.3 Review Questions: True/False

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

27.4 Review Questions: Multiple Choice

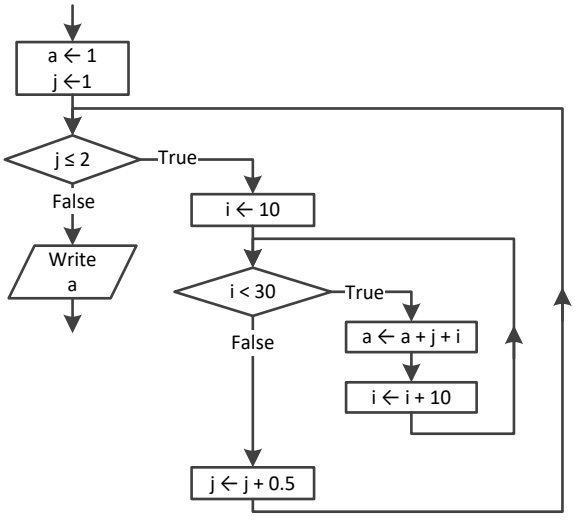
1. b
2. a
3. c
4. a
5. b

27.5 Review Exercises

1. Solution

i. 10
ii. A value greater than or equal to 4.5 and less than 5.0 ($4.5 \le x < 5.0$)
iii. −7 (or −8)
iv. 138 (or 139)

2. Solution



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

6	`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.WriteLine(a)`	It displays: 100		

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`		false	
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.WriteLine(s)`	It displays: 25		

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

4. Solution

For input value of “NO”

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

For input values of "YES", "NO"

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

16	`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.WriteLine(s)`	It displays: 76			

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.WriteLine(s)`	It displays: 46			

5. Solution

```
int hour, minutes;

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

6. Solution

```
int i, j;

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

7. Solution

```
int i, j;

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

8. Solution

```
int i, j;

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

9. Solution

```
int i, j, y;

Console.Write("Enter an integer between 3 and 20: ");
y = Convert.ToInt32(Console.ReadLine());

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

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

10. Solution

```
int i, j, y;

Console.Write("Enter an integer between 3 and 20: ");
y = Convert.ToInt32(Console.ReadLine());

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

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

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

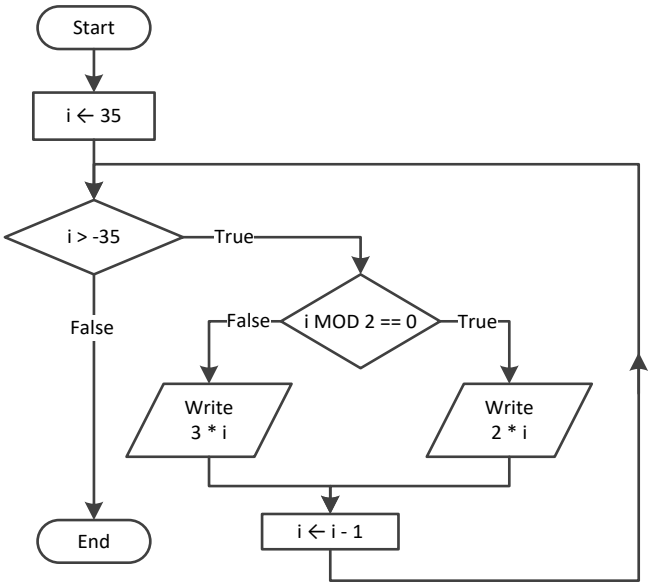
11. Solution

```
int i, j;

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

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

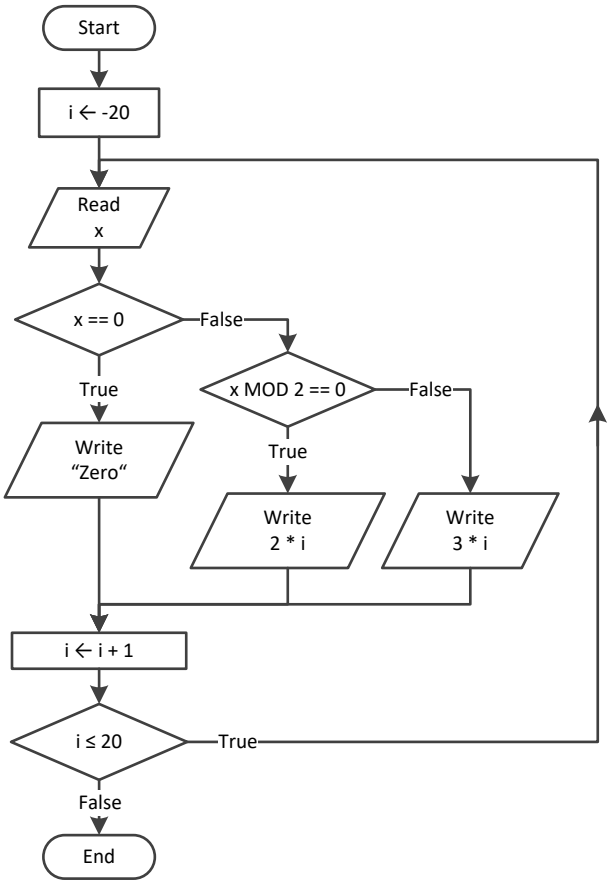
Chapter 28

28.4 Review Exercises

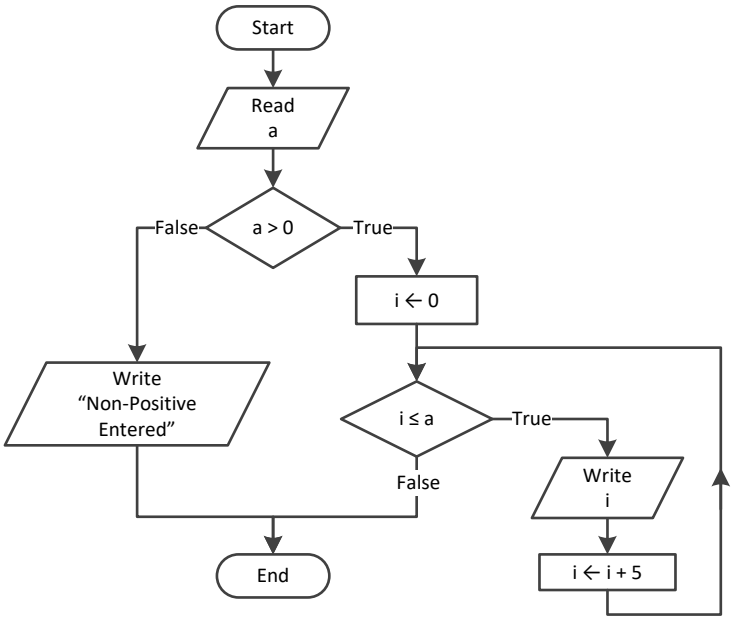
1. Solution



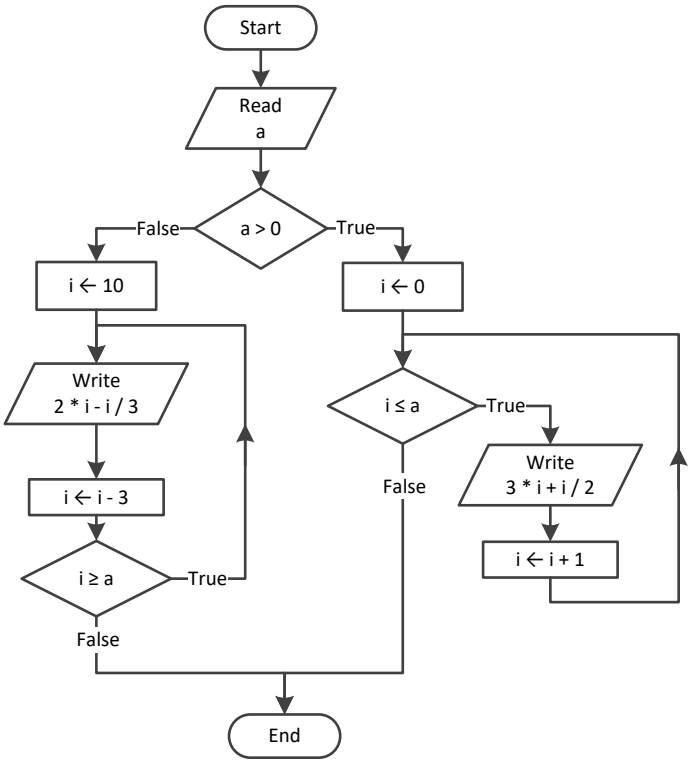
2. Solution



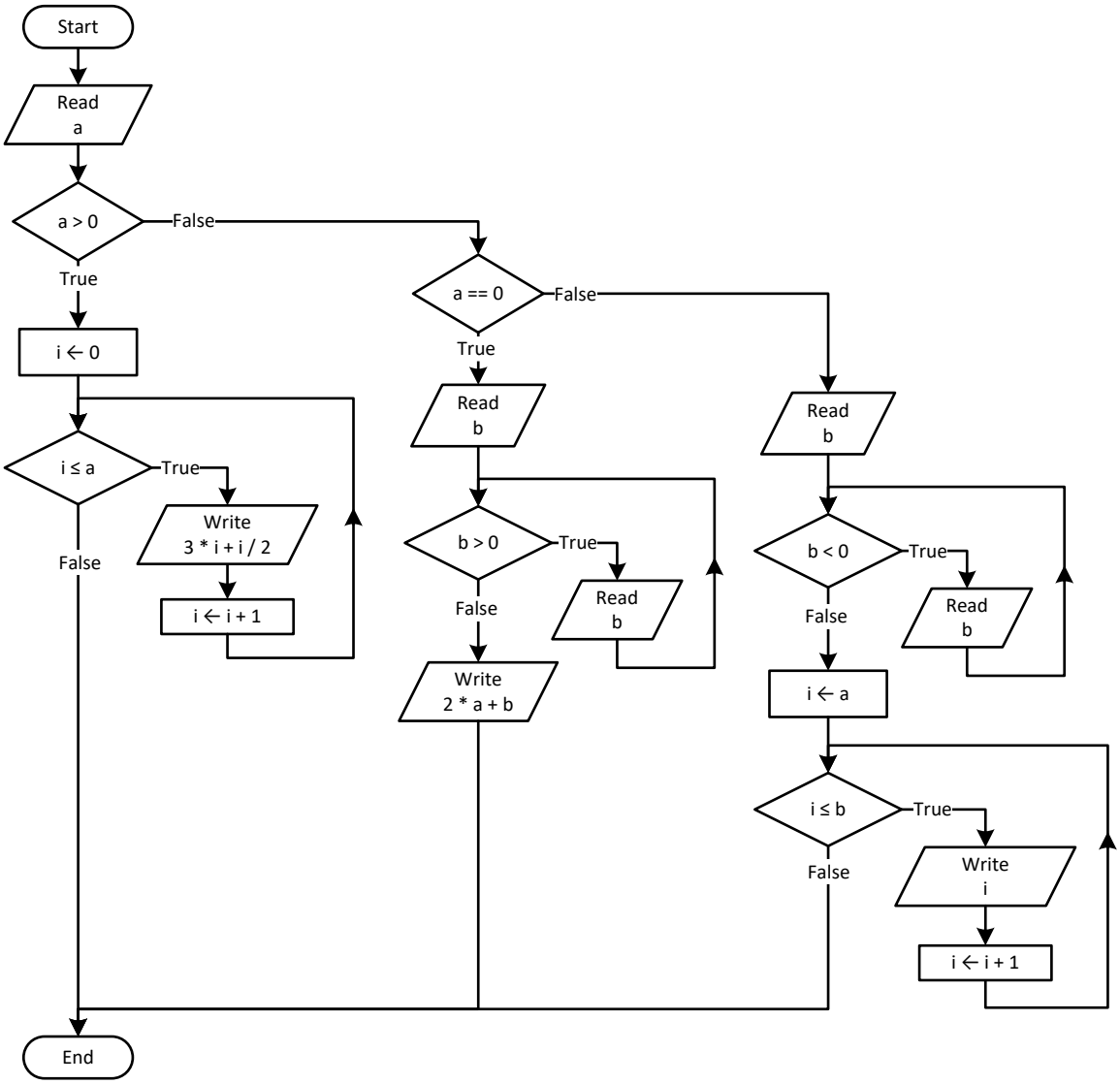
3. Solution



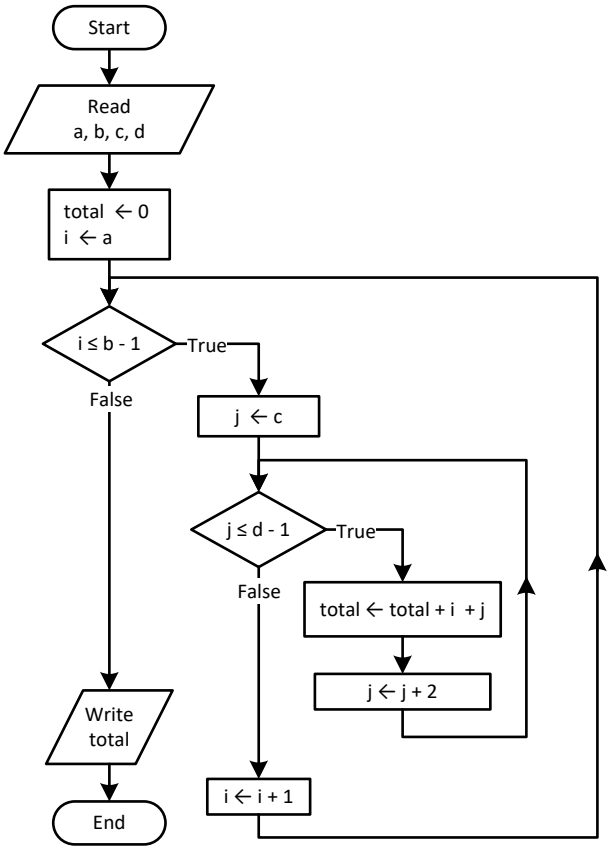
4. Solution


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

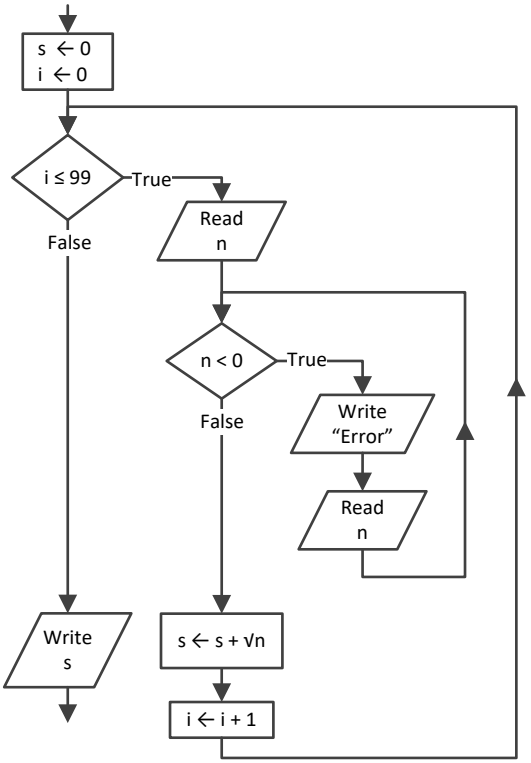
5. Solution


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

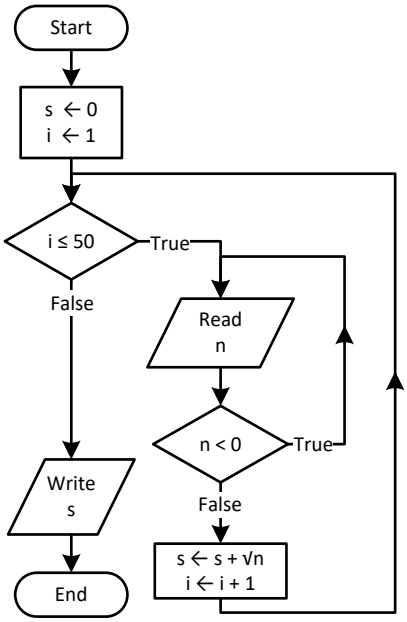
6. Solution


Start
Read
a, b, c, d
total ← 0
i ← a
i ≤ b - 1
True
False
j ← c
j ≤ d - 1
True
False
total ← total + i + j
j ← j + 2
Write
total
i ← i + 1
End

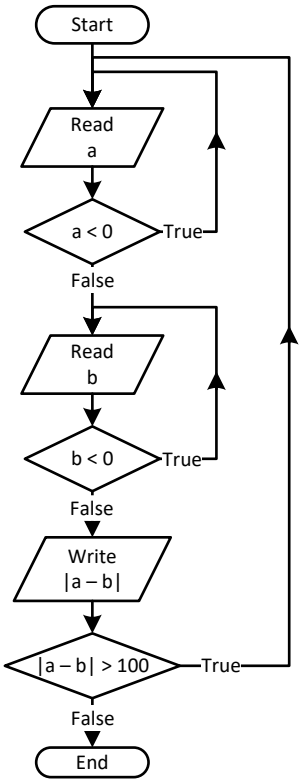
7. Solution



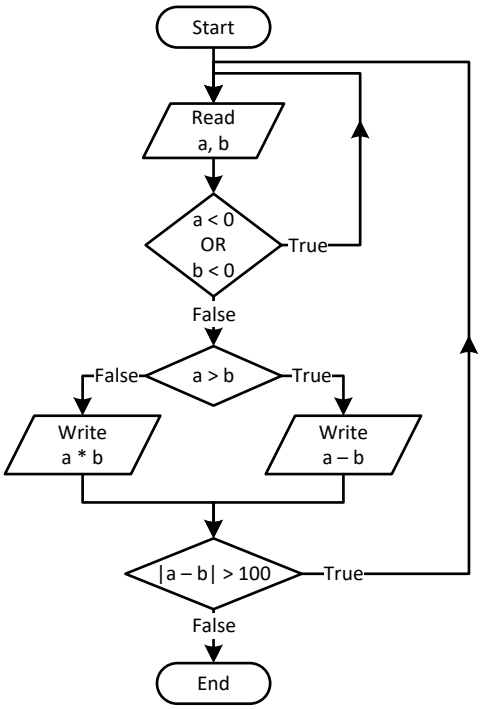
8. Solution



9. Solution



10. Solution



11. Solution

```
int a, i;

i = 0;
a = Convert.ToInt32(Console.ReadLine());
do {
  if (i % 2 != 0) {
    Console.WriteLine(i);
  }
  i += 5;
} while (i < a);
```

12. Solution

```
int a, b, i;

a = Convert.ToInt32(Console.ReadLine());
while (a != -1) {
  do {
    b = Convert.ToInt32(Console.ReadLine());
  } while (b <= a);
  for (i = a; i <= b; i++) {
    Console.WriteLine(i);
  }
  a = Convert.ToInt32(Console.ReadLine());
}
```

13. Solution

```
int i;
double P, S, a;

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

while (true) {
  if (i < 45) {
    S += a;
  }
  else {
    P *= a;
  }
  i++;
  if (i >= 90) break;
  a = Convert.ToDouble(Console.ReadLine());
}

Console.WriteLine(S + " " + P);
```

Chapter 29

29.8 Review Questions: True/False

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

29.9 Review Questions: Multiple Choice

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

29.10 Review Exercises

1. *Solution*

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

2. *Solution*

First approach

```
string text, character;
bool found;
int i;

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

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

```
  if (character == " ") {
    found = true;
    break;
  }
}

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

Second approach

```
string text;

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

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

3. Solution

First approach

```
string sentence, character;
bool found;
int i;
string digits = "0123456789";

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

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

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

Second approach

```
string sentence;
bool found;
int i;
string digit;
```

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

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

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

4. Solution

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

5. Solution

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

6. Solution

```
s = 0;
i = 1;
count = 100;
do {
  number = Convert.ToDouble(Console.ReadLine());
  s = s + number;
  i++;
} while (i <= count);
average = s / count;
Console.WriteLine(average);
```

7. Solution

```
int i, denom;
double s;

s = 0;
```

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

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

8. Solution

```
int i, j;

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

9. Solution

```
int i, j;

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

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

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

10. Solution

```
int i, j, n;

Console.Write("Enter an integer: ");
n = Convert.ToInt32(Console.ReadLine());

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

```
}
Console.WriteLine();

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

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

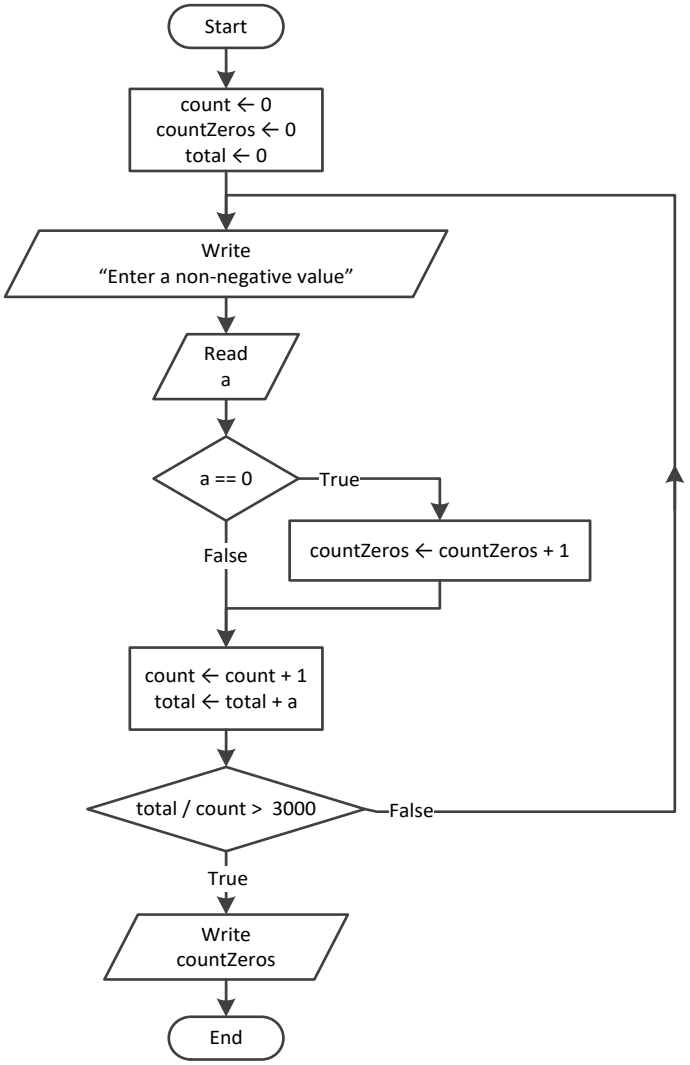
Chapter 30

30.7 Review Questions: True/False

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

30.8 Review Exercises

1. Solution




```
int count, countZeros;
double a, total;

count = 0;
countZeros = 0;
total = 0;
do {
  Console.Write("Enter a non-negative value: ");
```

```
  a = Convert.ToDouble(Console.ReadLine());
  if (a == 0) {
    countZeros++;
  }
  count++;
  total += a;
} while (total / count <= 3000);
Console.WriteLine(countZeros);
```

2. Solution

First approach

```
int a, d1, d2, d3, d4, i, r;

Console.Write("Enter an integer between 1 and 20: ");
a = Convert.ToInt32(Console.ReadLine());
for (i = 1000; i <= 9999; i++) {
  d4 = i % 10;
  r = (int)(i / 10);
  d3 = r % 10;
  r = (int)(r / 10);
  d2 = r % 10;
  d1 = (int)(r / 10);
  if (d1 + d2 + d3 + d4 < a) {
    Console.WriteLine(i);
  }
}
```

Second approach

```
int a, d1, d2, d3, d4;

Console.Write("Enter an integer between 1 and 20: ");
a = Convert.ToInt32(Console.ReadLine());
for (d1 = 1; d1 <= 9; d1++) {
  for (d2 = 0; d2 <= 9; d2++) {
    for (d3 = 0; d3 <= 9; d3++) {
      for (d4 = 0; d4 <= 9; d4++) {
        if (d1 + d2 + d3 + d4 < a) {
          Console.WriteLine(d1 * 1000 + d2 * 100 + d3 * 10 + d4);
        }
      }
    }
  }
}
```

3. Solution

First approach

```
int d1, d2, d3, d4, i, r;

for (i = 1000; i <= 9999; i++) {
  d4 = i % 10;
  r = (int)(i / 10);
```

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

Second approach

```
int d1, d2, d3, d4;

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

4. Solution

First approach

```
int x, count;

Console.Write("Enter an integer: ");
x = Convert.ToInt32(Console.ReadLine());

count = 0;

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

Console.WriteLine(count);
```

Second approach

```
int x, count;

Console.Write("Enter an integer: ");
x = Convert.ToInt32(Console.ReadLine());

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

Console.WriteLine(count);
```

5. Solution

```
x = Convert.ToInt32(Console.ReadLine());
while (x != 1 && x != 0) {
  Console.WriteLine("Error");
  x = Convert.ToInt32(Console.ReadLine());
}
```

6. Solution

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

7. Solution

```
int count;
double x, y;

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

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

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

8. Solution

```
string answer;
double area, r;

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

  area = Math.PI * Math.Pow(r, 2);
```

```
  Console.WriteLine("The area is: " + area);

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

9. Solution

```
int i;
double maximum, total, t;

maximum = -460;
total = 0;
for (i = 1; i <= 31; i++) {
  Console.Write("Enter temperature for day " + i + ": ");
  t = Convert.ToDouble(Console.ReadLine());
  while (t < -459.67) {
    Console.WriteLine("Error! Wrong temperature.");
    Console.Write("Enter temperature for day " + i + ": ");
    t = Convert.ToDouble(Console.ReadLine());
  }

  total += t;
  if (t > maximum) {
    maximum = t;
  }
}

Console.WriteLine(total / 31 + " " + maximum);
```

10. Solution

```
int hour, maxHour, maxMinutes, minHour, minMinutes, minutes;
double level, maximum, minimum;

level = Convert.ToDouble(Console.ReadLine());
if (level != 9999) {
  hour = Convert.ToInt32(Console.ReadLine());
  minutes = Convert.ToInt32(Console.ReadLine());

  maximum = level;
  maxHour = hour;
  maxMinutes = minutes;

  minimum = level;
  minHour = hour;
  minMinutes = minutes;

  level = Convert.ToDouble(Console.ReadLine());
  while (level != 9999) {
    hour = Convert.ToInt32(Console.ReadLine());
    minutes = Convert.ToInt32(Console.ReadLine());
```

```
    if (level > maximum) {
      maximum = level;
      maxHour = hour;
      maxMinutes = minutes;
    }

    if (level < minimum) {
      minimum = level;
      minHour = hour;
      minMinutes = minutes;
    }

    level = Convert.ToDouble(Console.ReadLine());
  }

  Console.WriteLine(maximum + ", " + maxHour + ", " + maxMinutes);
  Console.WriteLine(minimum + ", " + minHour + ", " + minMinutes);
}
```

11. Solution

```
string alphabet;
int number, total;

Console.Write("Enter an integer: ");
number = Convert.ToInt32(Console.ReadLine());

do {
  total = 0;
  while (number > 0) {
    total += number % 10;
    number = (int)(number / 10);
  }

  if (total > 26) {
    number = total;
  }
} while (total > 26);

alphabet = "ABCDEFGHIJKLMNOPQRSTUVWXYZ";
Console.Write("The name of the person who might be thinking ");
Console.WriteLine("of you starts with a(an): " + alphabet[total - 1]);
```

12. Solution

```
int x, y;

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

```
}
```

13. Solution

```
int x, y, z;

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

14. Solution

```
int m1, m2, m3, s;

m1 = Convert.ToInt32(Console.ReadLine());
m2 = Convert.ToInt32(Console.ReadLine());
m3 = Convert.ToInt32(Console.ReadLine());

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

m1 = s;
m2 = m3;

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

Console.WriteLine(s);
```

15. Solution

```
double a;
int x, numberOfDivisors, i;

a = Convert.ToDouble(Console.ReadLine());
```

```
while (a <= 0 || a != (int)a) {
  Console.WriteLine("Error! You must enter a positive integer");
  a = Convert.ToDouble(Console.ReadLine());
}
x = (int)a;

numberOfDivisors = 2;
for (i = 2; i <= (int)(x / 2); i++) {
  if (x % i == 0) {
    numberOfDivisors++;
  }
}
Console.WriteLine(numberOfDivisors);
```

16. Solution

```
int x, numberOfDivisors, i;

Console.Write("Enter an integer greater than 1: ");
x = Convert.ToInt32(Console.ReadLine());
while (x <= 1) {
  Console.WriteLine("Error!");
  Console.Write("Enter an integer greater than 1: ");
  x = Convert.ToInt32(Console.ReadLine());
}

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

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

17. Solution

```
int start, finish, c, x, y;
double z;

Console.Write("Enter an positive integer: ");
start = Convert.ToInt32(Console.ReadLine());
Console.Write("Enter a second positive integer: ");
finish = Convert.ToInt32(Console.ReadLine());

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

```
for (x = start; x <= finish; x++) {
  for (y = x; y <= finish; y++) {
    z = Math.Sqrt(x * x + y * y);
    //If result is integer and less than or equal to finish, display x, y, z
    if (z == (int)z && z <= finish) {
      Console.WriteLine(x + " " + y + " " + z);
    }
  }
}
```

18. Solution

```
int a, b, c, i, numberOfDivisors, x;

Console.Write("Enter an integer greater than 1: ");
a = Convert.ToInt32(Console.ReadLine());
while (a < 2) {
  Console.Write("Wrong number. Please enter an integer greater than 1: ");
  a = Convert.ToInt32(Console.ReadLine());
}

Console.Write("Enter a second integer greater than 1: ");
b = Convert.ToInt32(Console.ReadLine());
while (b < 2) {
  Console.Write("Wrong number. Please enter a second integer greater than 1: ");
  b = Convert.ToInt32(Console.ReadLine());
}

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

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

19. Solution

```
double y;
int x, i, total;
```

```
Console.Write("Enter a positive integer: ");
y = Convert.ToDouble(Console.ReadLine());
while (y <= 0 || y != (int)y) {
  Console.Write("Wrong number! Enter a positive integer: ");
  y = Convert.ToDouble(Console.ReadLine());
}
x = (int)y;

total = 0;
for (i = 1; i <= x - 1; i++) {
  if (x % i == 0) {
    total += i;
  }
}

if (total == x) {
  Console.WriteLine("Number " + x + " is a perfect number");
}
else {
  Console.WriteLine("Number " + x + " is not a perfect number");
}
```

20. Solution

```
double y;
int a, b, c, x, j, total;

Console.Write("Enter a positive integer: ");
y = Convert.ToDouble(Console.ReadLine());
while (y <= 0 || y != (int)y) {
  Console.Write("Wrong number! Enter a positive integer: ");
  y = Convert.ToDouble(Console.ReadLine());
}
a = (int)y;

Console.Write("Enter a second positive integer: ");
y = Convert.ToDouble(Console.ReadLine());
while (y <= 0 || y != (int)y) {
  Console.Write("Wrong number! Enter a second positive integer: ");
  y = Convert.ToDouble(Console.ReadLine());
}
b = (int)y;

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

for (x = a; x <= b + 1 - 1; x++) {
  total = 0;
  for (j = 1; j <= x - 1; j++) {
```

```
    if (x % j == 0) {
      total += j;
    }
  }

  if (total == x) {
    Console.WriteLine("Number " + x + " is a perfect number");
  }
}
```

21. Solution

```
int a, b, c, d1, d2, d3, d4, r, x;

Console.Write("Enter a positive four-digit integer: ");
a = Convert.ToInt32(Console.ReadLine());
while (a < 1000 || a > 9999) {
  Console.Write("Wrong number. Please enter a positive four-digit integer: ");
  a = Convert.ToInt32(Console.ReadLine());
}

Console.Write("Enter a second positive four-digit integer: ");
b = Convert.ToInt32(Console.ReadLine());
while (b < 1000 || b > 9999) {
  Console.Write("Wrong number. Please enter a second positive four-digit integer: ");
  b = Convert.ToInt32(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);
  }
}
```

22. Solution

```
int i;

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

```
}
```

23. Solution

```
int i, offset;

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

24. Solution

```
int i;

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

25. Solution

First approach

```
int i, offset, value;

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

Second approach

```
int i;
string value;

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

26. Solution

```
int a, fib, fibPrevious, fibPrevious2, i;

a = Convert.ToInt32(Console.ReadLine());

fibPrevious2 = 0;
fibPrevious = 1;
fib = 1;
for (i = 1; i <= a; i++) {
```

```
  Console.WriteLine(fibPrevious2);
  fib = fibPrevious + fibPrevious2;
  fibPrevious2 = fibPrevious;
  fibPrevious = fib;
}
```

27. Solution

```
int a, fib, fibPrev, fibPrevPrev;

a = Convert.ToInt32(Console.ReadLine());

fibPrevPrev = 0;
fibPrev = 1;
fib = 1;
while (fib < a) {
  Console.WriteLine(fib);
  fib = fibPrev + fibPrevPrev;
  fibPrevPrev = fibPrev;
  fibPrev = fib;
}
```

28. Solution

```
int denominator, i, n, nominator;
double y;

Console.Write("Enter a positive integer: ");
n = Convert.ToInt32(Console.ReadLine());
while (n <= 0) {
  Console.Write("Wrong number. Please enter a positive integer: ");
  n = Convert.ToInt32(Console.ReadLine());
}

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

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

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

29. Solution

```
int i, n, nominator, sign;
double y;

Console.Write("Enter a positive integer: ");
```

```
n = Convert.ToInt32(Console.ReadLine());
while (n <= 0) {
  Console.Write("Wrong number. Please enter a positive integer: ");
  n = Convert.ToInt32(Console.ReadLine());
}

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

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

30. Solution

```
int i, n, sign;
double y;

Console.Write("Enter an integer greater than 2: ");
n = Convert.ToInt32(Console.ReadLine());
while (n <= 2) {
  Console.Write("Wrong number. Please enter an integer greater than 2: ");
  n = Convert.ToInt32(Console.ReadLine());
}

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

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

Console.WriteLine(y);
```

31. Solution

```
int i, n;
double y;

Console.Write("Enter a positive integer: ");
n = Convert.ToInt32(Console.ReadLine());
while (n <= 0) {
  Console.Write("Wrong number. Please enter a positive integer: ");
  n = Convert.ToInt32(Console.ReadLine());
}

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

```
}

Console.WriteLine(y);
```

32. Solution

```
int factorial, i, n;

Console.Write("Enter a non-negative integer: ");
n = Convert.ToInt32(Console.ReadLine());

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

Console.WriteLine(factorial);
```

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

33. Solution

First approach

```
const double ACCURACY = 0.00001;

int i, j;
double factorial, exponentialPrevious, exponential, x;

x = Convert.ToDouble(Console.ReadLine());

exponential = 0;
i = 0;
do {
  exponentialPrevious = exponential;

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

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

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

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

Second approach

```
const double ACCURACY = 0.00001;

int i;
double factorial, exponentialPrevious, exponential, x;

x = Convert.ToDouble(Console.ReadLine());
```

```
exponential = 1;
i = 1;
factorial = 1;
do {
  exponentialPrevious = exponential;

  factorial *= i;

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

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

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

34. Solution

First approach

```
const double ACCURACY = 0.00001;

int i, j, sign;
double factorial;
double sinus, sinusPrevious, x;

x = Convert.ToDouble(Console.ReadLine());

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

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

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

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

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

Second approach

```
const double ACCURACY = 0.00001;

int i, sign;
double factorial;
double sinus, sinusPrevious, x;

x = Convert.ToDouble(Console.ReadLine());

sign = -1;
```

```
sinus = x;
i = 3;
factorial = 1;
do {
  sinusPrevious = sinus;

  factorial *= i * (i - 1);

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

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

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

35. Solution

First approach

```
const double ACCURACY = 0.00001;

int i, j, sign;
double factorial;
double cosinus, cosinusPrevious, x;

x = Convert.ToDouble(Console.ReadLine());

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

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

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

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

Console.WriteLine("cos(" + x + ") ~= " + cosinus);
```

Second approach

```
const double ACCURACY = 0.00001;

int i, sign;
double factorial;
double cosinus, cosinusPrevious, x;

x = Convert.ToDouble(Console.ReadLine());
```

```
sign = -1;
cosinus = 1;
i = 2;
factorial = 1;
do {
  cosinusPrevious = cosinus;

  factorial *= i * (i - 1);

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

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

Console.WriteLine("cos(" + x + ") ~= " + cosinus);
```

36. Solution

```
int a, b, c, i;
bool failure;

string alphabet = "abcdefghijklmnopqrstuvwxyz";

do {
  Console.Write("Enter an integer between 1 and 26: ");
  a = Convert.ToInt32(Console.ReadLine());

  failure = false;
  if (a < 1) {
    Console.WriteLine("Please enter positive integers!");
    failure = true;
  }
  else if (a > 26) {
    Console.WriteLine("Please enter a value less than or equal to 26!");
    failure = true;
  }
} while (failure);

do {
  Console.Write("Enter an integer between 1 and 26: ");
  b = Convert.ToInt32(Console.ReadLine());

  failure = false;
  if (b < 1) {
    Console.WriteLine("Please enter positive integers!");
    failure = true;
  }
  else if (b > 26) {
    Console.WriteLine("Please enter a value less than or equal to 26!");
    failure = true;
  }
} while (failure);
```

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

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

37. Solution

```
int attempts, guess, secretNumber;

Random rnd = new();

secretNumber = rnd.Next(1, 101);

attempts = 1;
Console.Write("Enter a guess: ");
guess = Convert.ToInt32(Console.ReadLine());
while (guess != secretNumber) {
  if (guess > secretNumber) {
    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 = Convert.ToInt32(Console.ReadLine());
}
Console.WriteLine("You found it!");
Console.WriteLine("Attempts: " + attempts);
```

38. Solution

```
int attempts = 0, firstPlayerAttempts = 0, guess, i, secretNumber;

Random rnd = new();

for (i = 1; i <= 2; i++) {
  secretNumber = rnd.Next(1, 101);

  attempts = 1;
  Console.Write("Enter a guess: ");
  guess = Convert.ToInt32(Console.ReadLine());
  while (guess != secretNumber) {
    if (guess > secretNumber) {
      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 = Convert.ToInt32(Console.ReadLine());
  }
  Console.WriteLine("You found it!");
  Console.WriteLine("Attempts: " + attempts);

  if (i == 1) {
    firstPlayerAttempts = attempts;
  }
}

if (firstPlayerAttempts < attempts) {
  Console.WriteLine("First player wins!");
}
else if (firstPlayerAttempts > attempts) {
  Console.WriteLine("Second player wins!");
}
else {
  Console.WriteLine("It's a draw");
}
```

39. Solution

```
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 = Convert.ToInt32(Console.ReadLine());

  if (choice == 1) {
    Console.Write("Enter diagonal: ");
    diagonal = Convert.ToInt32(Console.ReadLine());
    Console.WriteLine("Width: " + (diagonal * 0.8));
    Console.WriteLine("Height: " + (diagonal * 0.6));
  }
  else if (choice == 2) {
    Console.Write("Enter diagonal: ");
    diagonal = Convert.ToInt32(Console.ReadLine());
    Console.WriteLine("Width: " + (diagonal * 0.87));
    Console.WriteLine("Height: " + (diagonal * 0.49));
  }
} while (choice != 3);
```

40. Solution

```
int countA, countABoys, countB, countCdefGirls, grade;
```

```
int i, maximum, minimum, n, total, totalA, totalABoys, totalB;
string gender;

Console.Write("Enter total number of students: ");
n = Convert.ToInt32(Console.ReadLine());
while (n <= 0) {
  Console.Write("Wrong number. Please enter total number of students: ");
  n = Convert.ToInt32(Console.ReadLine());
}

total = 0;
totalA = 0;
countA = 0;
totalB = 0;
countB = 0;
totalABoys = 0;
countABoys = 0;
countCdefGirls = 0;

maximum = -1;
minimum = 101;

for (i = 1; i <= n; i++) {
  Console.Write("Enter grade for student No " + i + ": ");
  grade = Convert.ToInt32(Console.ReadLine());
  while (grade < 0 || grade > 100) {
    Console.Write("Wrong grade. Please enter grade for student No " + i + ": ");
    grade = Convert.ToInt32(Console.ReadLine());
  }

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

  if (grade >= 90 && grade <= 100) {
    totalA += grade;
    countA++;
    if (gender == "M") {
      totalABoys += grade;
      countABoys++;
    }
  }
  else if (grade >= 80 && grade <= 89) {
    totalB += grade;
    countB++;
  }
  else {
    if (gender == "F") {
      countCdefGirls++;
```

```
    }
  }

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

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

  total += grade;
}

if (countA > 0) {
  Console.Write("The average value of those who got an 'A' is: ");
  Console.WriteLine(totalA / (double)countA);
}
if (countB > 0) {
  Console.Write("The average value of those who got a 'B' is: ");
  Console.WriteLine(totalB / (double)countB);
}
if (countABoys > 0) {
  Console.Write("The average value of boys who got an 'A' is: ");
  Console.WriteLine(totalABoys / (double)countABoys);
}
Console.WriteLine("The total number of girls that got less than 'B' is: " + countCdefGirls);
Console.WriteLine("The highest grade is: " + maximum);
Console.WriteLine("The lowest grade is: " + minimum);
Console.WriteLine("The average grade of the whole class is: " + total / (double)n);
```

41. Solution

```
double amount, discount;
string answer;

do {
  Console.Write("Enter amount: ");
  amount = Convert.ToDouble(Console.ReadLine());
  while (amount <= 0) {
    Console.Write("Wrong amount. Please enter amount: ");
    amount = Convert.ToDouble(Console.ReadLine());
  }

  if (amount < 20) {
    discount = 0;
  }
  else if (amount < 50) {
    discount = 3;
  }
  else if (amount < 100) {
    discount = 5;
  }
```

```
  else {
    discount = 10;
  }

  Console.WriteLine("Discount: " + discount + "%");
  Console.WriteLine("Amount to pay (discount included): " + (amount - amount * discount / 100));

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

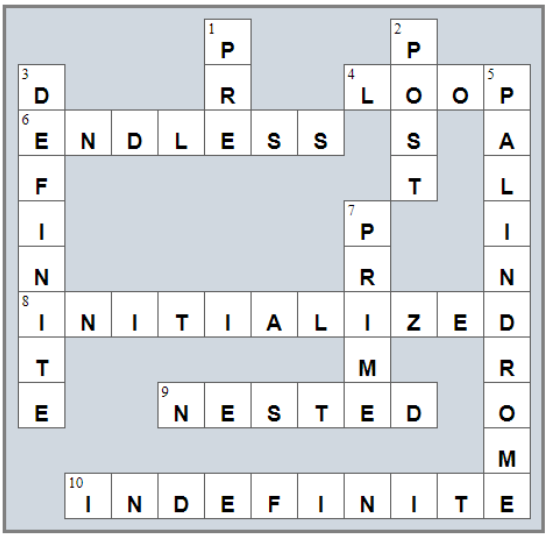
42. Solution

```
const double TAX_RATE = 0.25;

int kwh;
double t;

Console.Write("Enter number of Kilowatt-hours consumed: ");
kwh = Convert.ToInt32(Console.ReadLine());
while (kwh < 0 && kwh != -1) {
  Console.Write("Wrong value. Please enter number of Kilowatt-hours consumed: ");
  kwh = Convert.ToInt32(Console.ReadLine());
}

while (kwh != -1) {
  if (kwh <= 400) {
    t = kwh * 0.11;
  }
  else if (kwh <= 1500) {
    t = 400 * 0.11 + (kwh - 400) * 0.22;
  }
  else if (kwh <= 3500) {
    t = 400 * 0.11 + 1100 * 0.22 + (kwh - 1500) * 0.25;
  }
  else {
    t = 400 * 0.11 + 1100 * 0.22 + 2000 * 0.25 + (kwh - 3500) * 0.50;
  }

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

  Console.Write("Enter number of Kilowatt-hours consumed: ");
  kwh = Convert.ToInt32(Console.ReadLine());
  while (kwh < 0 && kwh != -1) {
    Console.Write("Wrong value. Please enter number of Kilowatt-hours consumed: ");
    kwh = Convert.ToInt32(Console.ReadLine());
  }
}
```

Review in “Loop Control Structures”

Review Crossword Puzzle

1.



Chapter 31

31.13 Review Questions: True/False

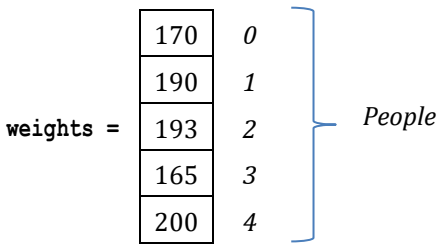
1. true
2. true
3. false
4. false
5. false
6. true
7. false
8. true
9. false
10. true
11. true
12. true
13. false
14. false
15. false
16. true
17. false
18. true
19. true
20. false
21. true
22. false
23. true
24. false
25. true
26. false
27. false
28. true
29. false
30. true
31. true
32. false
33. false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

31.14 Review Questions: Multiple Choice

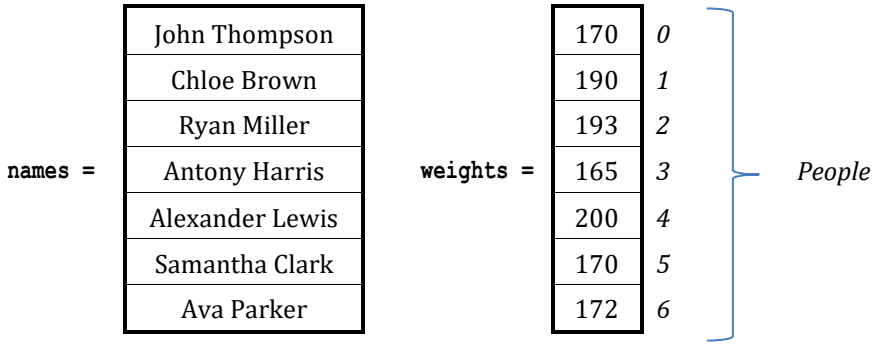
1. b
2. a
3. c
4. b
5. d
6. b
7. d
8. d
9. c
10. a
11. b
12. a
13. b
14. b

31.15 Review Exercises

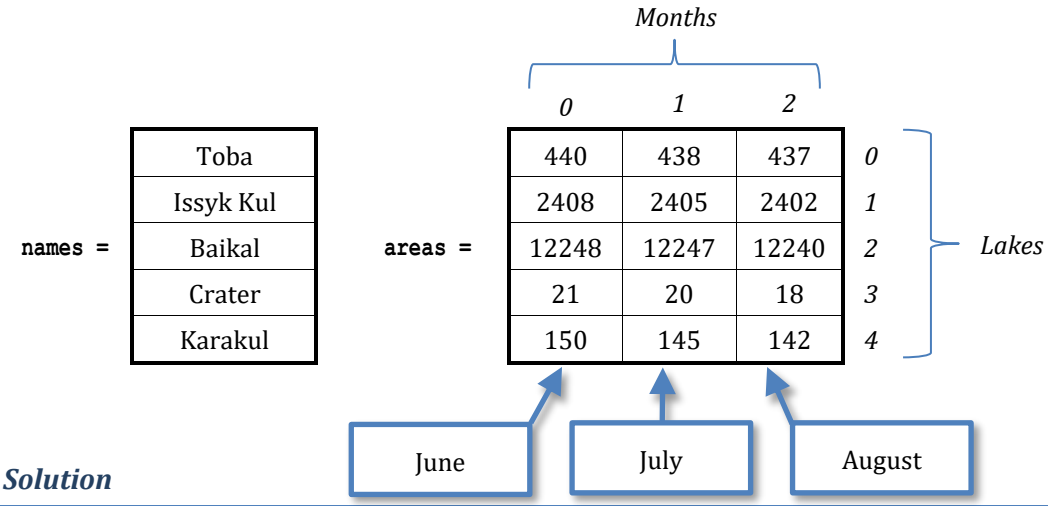
1. Solution



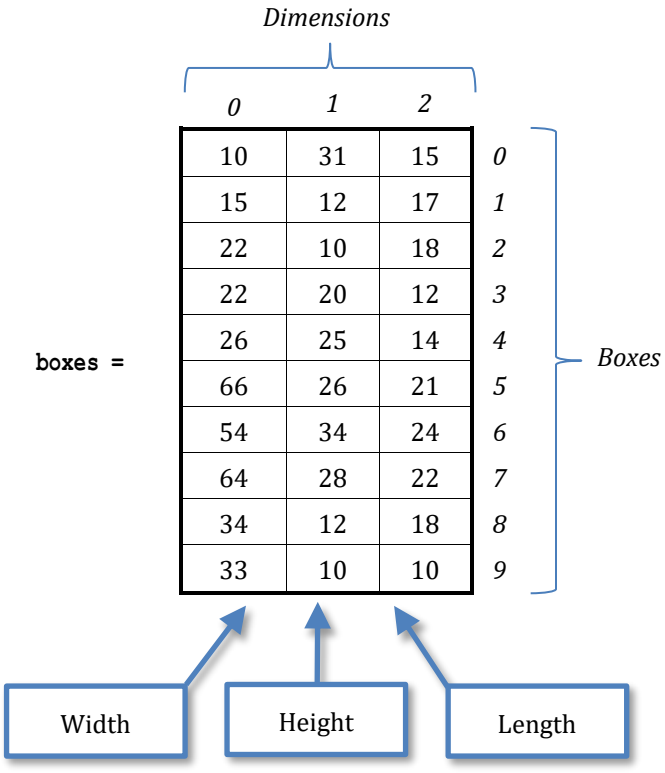
2. Solution



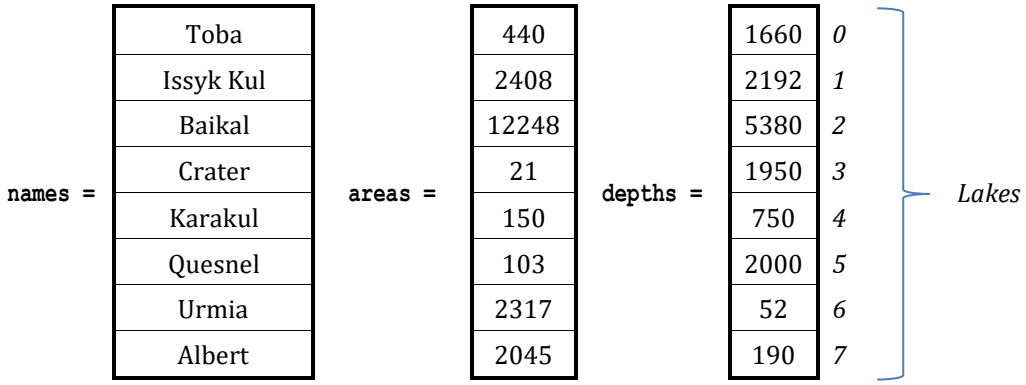
3. Solution



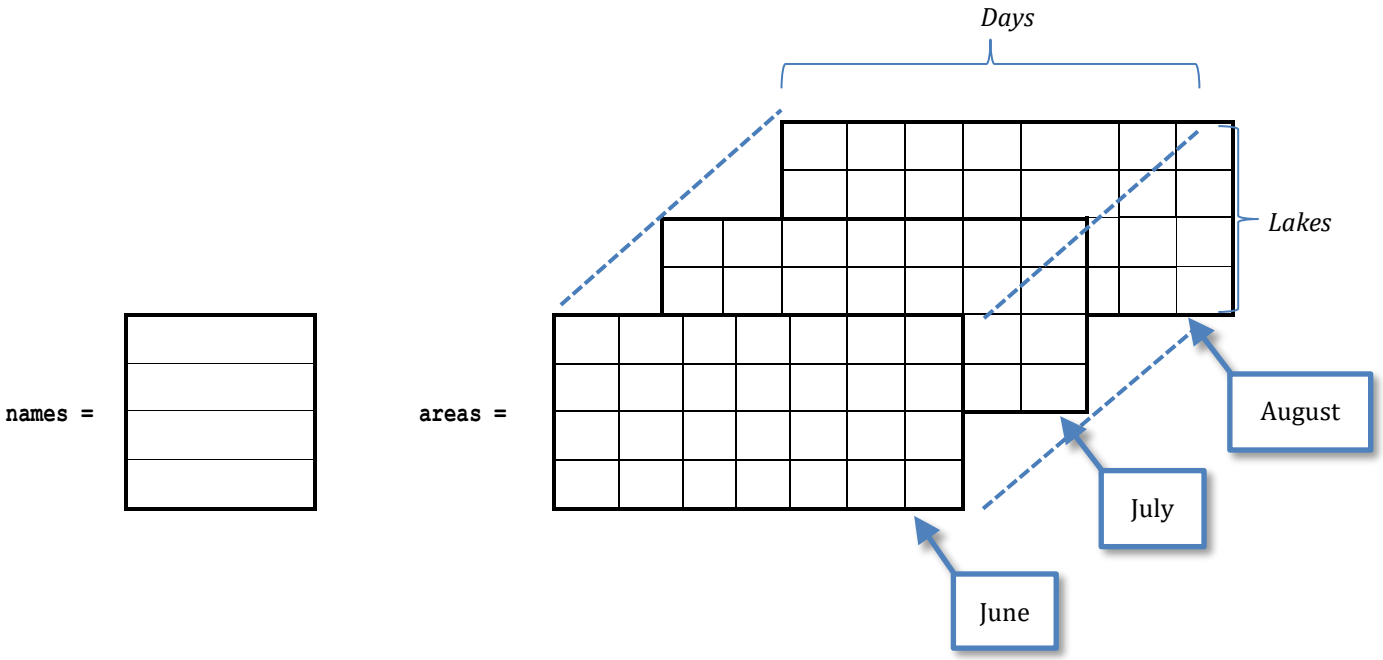
4. Solution



5. Solution



6. Solution



7. Solution

Step	Statement	x	a[0]	a[1]	a[2]
1	`int[] a = new int[3]`	?	?	?	?
2	`a[2] = 1`	?	?	?	**1**
3	`x = 0`	**0**	?	?	1
4	`a[x + a[2]] = 4`	0	?	**4**	1
5	`a[x] = a[x + 1] * 4`	0	**16**	4	1

8. Solution

Step	Statement	x	a[0]	a[1]	a[2]	a[3]	a[4]
1	int[] a = new int[5]	?	?	?	?	?	?
2	a[1] = 5	?	?	**5**	?	?	?
3	x = 0	**0**	?	5	?	?	?
4	a[x] = 4	0	**4**	5	?	?	?
5	a[a[0]] = a[x + 1] % 3	0	4	5	?	?	**2**
6	a[a[0] / 2] = 10	0	4	5	**10**	?	2
7	x += 2	**2**	4	5	10	?	2
8	a[x + 1] = a[x] + 9	2	4	5	10	**19**	2

9. Solution

For input value of 3

Step	Statement	x	a[0]	a[1]	a[2]	a[3]
1	int[] a = new int[4]	?	?	?	?	?
2	a[1] = Convert.ToInt32(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] = Convert.ToInt32(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] = Convert.ToInt32(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

10. 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] = Convert.ToInt32(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] = Convert.ToInt32(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] = Convert.ToInt32(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

11. Solution

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

4, 5	`if (x > y)`
	`  a[0] = 1;`
	`else`
	`  a[0] = y;`
		4	3	**1**	?	?

6	`a[1] = x + 3`	4	3	1	**7**	?
7	`y = y - 1`	4	**2**	1	7	?
8	`a[y] = (x + 5) % 2`	4	2	1	7	**1**

12. Solution

Step	Statement	i	a[0]	a[1]	a[2]	a[3]	a[4]	a[5]
1	`int[] a = {17, 12, 45, 12, 12, 49}`	?	**17**	**12**	**45**	**12**	**12**	**49**
2	`i = 0`	**0**	17	12	45	12	12	49
3	`i <= 5`	true						
4	`if (a[i] == 12)`	false						
5	`a[i]++`	0	**18**	12	45	12	12	49
6	`i++`	**1**	18	12	45	12	12	49
7	`i <= 5`	true						
8	`if (a[i] == 12)`	true						
9	`a[i]--`	1	18	**11**	45	12	12	49
10	`i++`	**2**	18	11	45	12	12	49
11	`i <= 5`	true						
12	`if (a[i] == 12)`	false						
13	`a[i]++`	2	18	11	**46**	12	12	49
14	`i++`	**3**	18	11	46	12	12	49
15	`i <= 5`	true						
16	`if (a[i] == 12)`	true						
17	`a[i]--`	3	18	11	46	**11**	12	49
18	`i++`	**4**	18	11	46	11	12	49
19	`i <= 5`	true						
20	`if (a[i] == 12)`	true						
21	`a[i]--`	4	18	11	46	11	**11**	49
22	`i++`	**5**	18	11	46	11	11	49
23	`i <= 5`	true						

24	`if (a[i] == 12)`	false						
25	`a[i]++`	5	18	11	46	11	11	**50**
26	`i++`	**6**	18	11	46	11	11	50
27	`i <= 5`	false						

13. Solution

Step	Statement	i	a[0]	a[1]	a[2]	a[3]	a[4]	a[5]
1	`int[] a = {10, 15, 12, 23, 22, 19}`	?	**10**	**15**	**12**	**23**	**22**	**19**
2	`i = 1`	**1**	10	15	12	23	22	19
3	`i <= 4`	true						
4	`a[i] = a[i + 1] + a[i - 1]`	1	10	**22**	12	23	22	19
5	`i++`	**2**	10	22	12	23	22	19
6	`i <= 4`	true						
7	`a[i] = a[i + 1] + a[i - 1]`	2	10	22	**45**	23	22	19
8	`i++`	**3**	10	22	45	23	22	19
9	`i <= 4`	true						
10	`a[i] = a[i + 1] + a[i - 1]`	3	10	22	45	**67**	22	19
11	`i++`	**4**	10	22	45	67	22	19
12	`i <= 4`	true						
13	`a[i] = a[i + 1] + a[i - 1]`	4	10	22	45	67	**86**	19
14	`i++`	**5**	10	22	45	67	86	19
15	`i <= 4`	false						

14. Solution

It displays:
Navajo
Cherokee
Sioux

15. Solution

```
const int ELEMENTS = 100;

int i;

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

for (i = 0; i <= ELEMENTS - 1; i++) {
  Console.WriteLine(Math.Pow(a[i], 3));
}
```

16. Solution

```
const int ELEMENTS = 80;

int i;

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

for (i = 0; i <= ELEMENTS - 1; i++) {
  a[i] = Math.Pow(a[i], 2);
}

for (i = ELEMENTS - 1; i >= 0; i--) {
  Console.WriteLine(a[i]);
}
```

17. Solution

```
const int ELEMENTS = 90;

int i;

int[] a = new int[ELEMENTS];
for (i = 0; i <= ELEMENTS - 1; i++) {
  a[i] = Convert.ToInt32(Console.ReadLine());
}

for (i = ELEMENTS - 1; i >= 0; i--) {
  if (a[i] % 5 == 0) {
    Console.WriteLine(a[i]);
  }
}
```

18. Solution

```
const int ELEMENTS = 50;

int i;

int[] a = new int[ELEMENTS];
for (i = 0; i <= ELEMENTS - 1; i++) {
  a[i] = Convert.ToInt32(Console.ReadLine());
}

for (i = 0; i <= ELEMENTS - 1; i++) {
  if (a[i] % 2 == 0 || a[i] > 10) {
    Console.WriteLine(a[i]);
  }
}
```

19. Solution

```
const int ELEMENTS = 30;

int i;
double total;

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

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

20. Solution

```
const int ELEMENTS = 50;

int i, total;

int[] a = new int[ELEMENTS];
for (i = 0; i <= ELEMENTS - 1; i++) {
  a[i] = Convert.ToInt32(Console.ReadLine());
}

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

21. Solution

```
const int ELEMENTS = 40;

int i;
double sumNeg, sumPos;

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

sumPos = 0;
sumNeg = 0;
for (i = 0; i <= ELEMENTS - 1; i++) {
```

```
  if (a[i] > 0 ) {
    sumPos += a[i];
  }
  else if (a[i] < 0) {
    sumNeg += a[i];
  }
}
Console.WriteLine(sumPos + ", " + sumNeg);
```

22. Solution

```
const int ELEMENTS = 20;

int i;
double total;

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

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

23. Solution

```
const int ELEMENTS = 50;

int i;

int[] a = new int[ELEMENTS];
for (i = 0; i <= ELEMENTS - 1; i++) {
  Console.Write("Enter an integer: ");
  a[i] = Convert.ToInt32(Console.ReadLine());
}

for (i = 0; i <= ELEMENTS - 1; i++) {
  if (a[i] < 20) {
    Console.WriteLine(a[i]);
  }
}
```

24. Solution

```
const int ELEMENTS = 60;

int i;

double[] a = new double[ELEMENTS];
for (i = 0; i <= ELEMENTS - 1; i++) {
  Console.Write("Enter a number: ");
  a[i] = Convert.ToDouble(Console.ReadLine());
}
```

```
for (i = 0; i <= ELEMENTS - 1; i += 2) {
  Console.WriteLine(a[i]);
}
```

25. Solution

```
const int ELEMENTS = 20;

int i;
double total;

double[] a = new double[ELEMENTS];
for (i = 0; i <= ELEMENTS - 1; i++) {
  Console.Write("Enter a number: ");
  a[i] = Convert.ToDouble(Console.ReadLine());
}

total = 0;
for (i = 0; i <= ELEMENTS - 1; i += 2) {
  total += a[i];
}
Console.WriteLine(total);
```

26. Solution

```
const int ELEMENTS = 100;

int i;
int[] a = new int[ELEMENTS];
for (i = 0; i <= ELEMENTS - 1; i++) {
  a[i] = i + 1;
}
…
```

27. Solution

First approach

```
const int ELEMENTS = 100;

int i, k;
int[] a = new int[ELEMENTS];
k = 2;
for (i = 0; i <= ELEMENTS - 1; i++) {
  a[i] = k;
  k += 2;
}
```

Second approach

```
const int ELEMENTS = 100;

int i;
int[] a = new int[ELEMENTS];
for (i = 0; i <= ELEMENTS - 1; i++) {
  a[i] = (i + 1) * 2;
}
```

28. Solution

```
int i, n;

Console.Write("Enter N: ");
n = Convert.ToInt32(Console.ReadLine());
while (n < 1) {
  Console.WriteLine("Error! Value must be greater than or equal to 1");
  Console.Write("Enter N: ");
  n = Convert.ToInt32(Console.ReadLine());
}

int[] a = new int[n];
for (i = 1; i <= n; i++) {
  a[i - 1] = (int)Math.Pow(i, 2);
}

for (i = 0; i <= n - 1; i++) {
  Console.WriteLine(a[i]);
}
```

29. Solution

```
const int ELEMENTS = 10;

int i;

double[] a = new double[ELEMENTS];
for (i = 1; i <= ELEMENTS - 1; i++) {
  Console.Write("Enter a number: ");
  a[i] = Convert.ToDouble(Console.ReadLine());
}

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

30. Solution

```
const int ELEMENTS = 50;

int i, count;

double[] a = new double[ELEMENTS];
for (i = 1; i <= ELEMENTS - 1; i++) {
  Console.Write("Enter a number: ");
  a[i] = Convert.ToDouble(Console.ReadLine());
}

count = 0;
for (i = 0; i <= ELEMENTS - 1; i++) {
  if (a[i] < 0) {
    count++;
  }
```

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

31. Solution

```
const int WORDS = 50;

int i;

string[] a = new string[WORDS];
for (i = 0; i <= WORDS - 1; i++) {
  a[i] = Console.ReadLine();
}

for (i = 0; i <= WORDS - 1; i++) {
  if (a[i].Length >= 10 ) {
    Console.WriteLine(a[i]);
  }
}
```

32. Solution

```
const int ELEMENTS = 30;

int i, k;

string[] words = new string[ELEMENTS];
for (i = 0; i <= ELEMENTS - 1; i++) {
  words[i] = Console.ReadLine();
}

int[] lengthLimits = {0, 5, 10, 20};

for (k = 1; k <= 3; k++) {
  for (i = 0; i <= ELEMENTS - 1; i++) {
    if (words[i].Length >= lengthLimits[k - 1] && words[i].Length < lengthLimits[k]) {
      Console.WriteLine(words[i]);
    }
  }
}
```

33. Solution

```
const int WORDS = 40;

int count, i, j;

string[] a = new string[WORDS];
for (i = 0; i <= WORDS - 1; i++) {
  Console.Write("Enter a word: ");
  a[i] = Console.ReadLine();
}

for (i = 0; i <= WORDS - 1; i++) {
  count = 0;
```

```
  for (j = 0; j <= a[i].Length - 1; j++) {
    if (a[i].Substring(j, 1) == "w") {
      count++;
    }
  }
  if (count >= 2) {
    Console.WriteLine(a[i]);
  }
}
```

34. Solution

```
string roman;
int number, digit1, digit2;
Console.Write("Enter a number between 1 and 99: ");
number = Convert.ToInt32(Console.ReadLine());

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

Dictionary<int, string> number2romanOnes = new() {
  {1, "I"}, {2, "II"}, {3, "III"}, {4, "IV"}, {5, "V"}, {6, "VI"}, {7, "VII"}, {8, "VIII"}, {9, "IX"}
};

Dictionary<int, string> number2romanTens = new() {
  {1, "X"}, {2, "XX"}, {3, "XXX"}, {4, "XL"}, {5, "L"}, {6, "LX"}, {7, "LXX"}, {8, "LXXX"}, {9, "XC"}
};

roman = number2romanTens[digit1] + number2romanOnes[digit2];
Console.WriteLine(roman);
```

Chapter 32

32.7 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

32.8 Review Questions: Multiple Choice

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

32.9 Review Exercises

1. Solution

Step	Statement	x	a
1	`int[,] a = new int[2, 3]`	?	? ? ? ? ? ?
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 7 10 11 ?

24	`j <= 2`	true		
25	`a[i, j] = (i + 1) * 5 + j`	1	2	5 6 7 10 11 **12**
26	`j++`	1	**3**	5 6 7 10 11 12
27	`j <= 2`	false		
28	`i++`	**2**	3	5 6 7 10 11 12
29	`i <= 1`	false		

3. Solution

Step	Statement	i	j	a
1	`int[,] a = new int[3, 3]`	?	?	? ? ? ? ? ? ? ? ?
2	`j = 0`	?	**0**	? ? ? ? ? ? ? ? ?
3	`j <= 2`	True		
4	`i = 0`	0	**0**	? ? ? ? ? ? ? ? ?
5	`i <= 2`	True		
6	`a[i, j] = (i + 1) * 2 + j * 4`	0	0	**2** ? ? ? ? ? ? ? ?
7	`i++`	**1**	0	2 ? ? ? ? ? ? ? ?
8	`i <= 2`	True		
9	`a[i, j] = (i + 1) * 2 + j * 4`	1	0	2 ? ? **4** ? ? ? ? ?

10	`i++`	**2**	0	2 ? ? 4 ? ? ? ? ?
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;

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] = Convert.ToInt32(Console.ReadLine());
  }
}

for (i = 0; i <= ROWS - 1; i++) {
  for (j = 0; j <= COLUMNS - 1; j++) {
    if (a[i, j] % 2 != 0) {
      Console.WriteLine(i + ", " + j);
    }
  }
}
```

10. Solution

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

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] = Convert.ToDouble(Console.ReadLine());
  }
}

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

11. Solution

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

```
int i, j;
double total;

double[,] a = new double[ROWS, COLUMNS];
for (i = 0; i <= ROWS - 1; i++) {
  for (j = 0; j <= COLUMNS - 1; j++) {
    a[i, j] = Convert.ToDouble(Console.ReadLine());
  }
}

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

12. Solution

```
const int N = 8 ;

int i, j, k;
double sumAntidiagonal, sumDiagonal;

double[,] a = new double[N, N];
for (i = 0; i <= N - 1; i++) {
  for (j = 0; j <= N - 1; j++) {
    a[i, j] = Convert.ToDouble(Console.ReadLine());
  }
}

sumDiagonal = 0;
sumAntidiagonal = 0;
for (k = 0; k <= N - 1; k++) {
  sumDiagonal += a[k, k];
  sumAntidiagonal += a[k, N - k - 1];
}
Console.WriteLine(sumDiagonal / N + ", " + sumAntidiagonal / N);
```

13. Solution

```
const int N = 5;

int i, j;

int[,] a = new int[N, N];
for (i = 0; i <= N - 1; i++) {
  for (j = 0; j <= N - 1; j++) {
    if (i == N - j - 1) {
      a[i, j] = 5;
    }
    else if (i > N - j - 1) {
```

```
      a[i, j] = 88;
    }
    else {
      a[i, j] = 11;
    }
  }
}

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

14. Solution

```
const int N = 5;

int i, j;

int[,] a = new int[N, N];
for (i = 0; i <= N - 1; i++) {
  for (j = 0; j <= N - 1; j++) {
    if (i == N - j - 1) {
      a[i, j] = 5;
    }
    else if (i > N - j - 1) {
      a[i, j] = 88;
    }
    else {
      a[i, j] = 11;
    }
    if (i == j) {
      a[i, j] = 0;
    }
  }
}

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

15. Solution

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

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] = Convert.ToDouble(Console.ReadLine());
  }
}

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

16. Solution

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

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] = Convert.ToDouble(Console.ReadLine());
  }
}

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

17. Solution

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

int i, j;

string[,] a = new string[ROWS, COLUMNS];
for (i = 0; i <= ROWS - 1; i++) {
  for (j = 0; j <= COLUMNS - 1; j++) {
    a[i, j] = Console.ReadLine();
  }
}
```

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

18. Solution

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

int i, j;

string[,] a = new string[ROWS, COLUMNS];
for (i = 0; i <= ROWS - 1; i++) {
  for (j = 0; j <= COLUMNS - 1; j++) {
    a[i, j] = Console.ReadLine();
  }
}

for (i = 0; i <= ROWS - 1; i++) {
  for (j = 0; j <= COLUMNS - 1; j++) {
    if (a[i, j].Length < 5) {
      Console.WriteLine(a[i, j]);
    }
  }
}
```

19. Solution

First approach

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

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[] lengthLimits = {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 < lengthLimits[k]) {
        Console.WriteLine(a[i, j]);
      }
    }
  }
```

```
}
```

Second approach

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

int i, j, k;

string[,] a = new string[ROWS,  COLUMNS];
for (i = 0; i <= ROWS - 1; i++) {
  for (j = 0; j <= COLUMNS - 1; j++) {
    a[i, j] = Console.ReadLine();
  }
}

for (k = 0; k <= 2; k++) {
  for (i = 0; i <= ROWS - 1; i++) {
    for (j = 0; j <= COLUMNS - 1; j++) {
      if (a[i, j].Length < 5 * Math.Pow(2, k)) {
        Console.WriteLine(a[i, j]);
      }
    }
  }
}
```

Chapter 33

33.8 Review Questions: True/False

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

33.9 Review Questions: Multiple Choice

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

33.10 Review Exercises

1. Solution

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

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] = Convert.ToInt32(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");
  }
}
```

2. Solution

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

int i, j, total;

int[,] g = new int[OBJECTS, FALLS];
for (i = 0; i <= OBJECTS - 1; i++) {
  for (j = 0; j <= FALLS - 1; j++) {
    g[i, j] = Convert.ToInt32(Console.ReadLine());
  }
}

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

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

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

3. Solution

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

int i, j, total;

int[,] points = new int[PLAYERS, MATCHES];
for (i = 0; i <= PLAYERS - 1; i++) {
  for (j = 0; j <= MATCHES - 1; j++) {
    points[i, j] = Convert.ToInt32(Console.ReadLine());
  }
}

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

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

4. Solution

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

int i, j;
double total;

double[,] temperatures = new double[CITIES, HOURS];
for (i = 0; i <= CITIES - 1; i++) {
  for (j = 0; j <= HOURS - 1; j++) {
    temperatures[i, j] = Convert.ToInt32(Console.ReadLine());
  }
}

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

```
}
```

5. Solution

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

int i, j, total;

string[] names = new string[PLAYERS];
int[,] goals = new int[PLAYERS, MATCHES];
for (i = 0; i <= PLAYERS - 1; i++) {
  names[i] = Console.ReadLine();
  for (j = 0; j <= MATCHES - 1; j++) {
    goals[i, j] = Convert.ToInt32(Console.ReadLine());
  }
}

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

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

6. Solution

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

int i, j, total;

string[] names = new string[STUDENTS];
int[,] grades = new int[STUDENTS, LESSONS];
for (i = 0; i <= STUDENTS - 1; i++) {
  names[i] = Console.ReadLine();
  for (j = 0; j <= LESSONS - 1; j++) {
    grades[i, j] = Convert.ToInt32(Console.ReadLine());
  }
}

double[] average = new double[STUDENTS];
for (i = 0; i <= STUDENTS - 1; i++) {
  total = 0;
```

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

for (j = 0; j <= LESSONS - 1; j++) {
  total = 0;
  for (i = 0; i <= STUDENTS - 1; i++) {
    total += grades[i, j];
  }
  Console.WriteLine(total / (double)STUDENTS);
}

for (i = 0; i <= STUDENTS - 1; i++) {
  if (average[i] < 60) {
    Console.WriteLine(names[i]);
  }
}

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

7. Solution

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

int i, j, total;

string[] judgeNames = new string[JUDGES];
for (j = 0; j <= JUDGES - 1; j++) {
  Console.Write("Enter name for judge No " + (j + 1) + ": ");
  judgeNames[j] = Console.ReadLine();
}

string[] artistNames = new string[ARTISTS];
string[] songTitles = 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) + ": ");
  artistNames[i] = Console.ReadLine();
  Console.Write("Enter song title for artist " + artistNames[i] + ": ");
  songTitles[i] = Console.ReadLine();
  for (j = 0; j <= JUDGES - 1; j++) {
    Console.Write("Enter score for artist: " + artistNames[i]);
    Console.Write(" gotten from judge " + judgeNames[j] + ": ");
    score[i, j] = Convert.ToInt32(Console.ReadLine());
  }
```

```
}

for (i = 0; i <= ARTISTS - 1; i++) {
  total = 0;
  for (j = 0; j <= JUDGES - 1; j++) {
    total += score[i, j];
  }
  Console.WriteLine(artistNames[i] + ", " + songTitles[i] + ": " + total);
}

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

8. Solution

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

int i, j, sumHeights, sumWeights;
double averageHeight, averageWeight;

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] = Convert.ToInt32(Console.ReadLine());
    heights[i, j] = Convert.ToInt32(Console.ReadLine());
  }
}

for (i = 0; i <= PEOPLE - 1; i++) {
  sumWeights = 0;
  sumHeights = 0;
  for (j = 0; j <= MONTHS - 1; j++) {
    sumWeights += weights[i, j];
    sumHeights += heights[i, j];
  }
  averageWeight = sumWeights / (double)MONTHS;
  averageHeight = sumHeights / (double)MONTHS;
  Console.WriteLine(averageWeight + ", " + averageHeight);
  Console.WriteLine(averageWeight * 702 / Math.Pow(averageHeight, 2));
}

for (i = 0; i <= PEOPLE - 1; i++) {
  Console.WriteLine(weights[i, 4] * 702 / Math.Pow(heights[i, 4], 2));
  Console.WriteLine(weights[i, 7] * 702 / Math.Pow(heights[i, 7], 2));
}
```

9. Solution

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

int consumed, i;
double payment, total;

int[,] meterReading = new int[CONSUMERS, 2];
for (i = 0; i <= CONSUMERS - 1; i++) {
  meterReading[i, 0] = Convert.ToInt32(Console.ReadLine());
  meterReading[i, 1] = Convert.ToInt32(Console.ReadLine());
}

total = 0;
for (i = 0; i <= CONSUMERS - 1; i++) {
  consumed = meterReading[i, 1] - meterReading[i, 0];
  Console.WriteLine(consumed);
  payment = consumed * 0.07;
  payment += VAT * payment;
  Console.WriteLine(payment);

  total += consumed;
}

Console.WriteLine(total + ", " + (total * 0.07 + total * 0.07 * VAT));
```

10. Solution

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

int i, j;
double average, total, usd;

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

string[] currency = {"British Pounds Sterling", "Euros", "Canadian Dollars", "Australian Dollars"};

double[,] rate = {
  {1.420, 1.421, 1.432, 1.431, 1.441},
  {1.043, 1.056, 1.038, 1.022, 1.029},
  {0.757, 0.764, 0.760, 0.750, 0.749},
  {0.620, 0.625, 0.629, 0.636, 0.639}
};

for (i = 0; i <= CURRENCIES - 1; i++) {
  total = 0;
  for (j = 0; j <= DAYS - 1; j++) {
    total += rate[i, j];
  }
  average = total / DAYS;
```

```
  Console.WriteLine(usd + " US dollars = " + (usd / average) + " " + currency[i]);
}
```

11. Solution

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

int i, j;
double totalGrossPay, grossPay, payRate, total;

string[] days = {"Monday", "Tuesday", "Wednesday", "Thursday", "Friday"};

payRate = Convert.ToDouble(Console.ReadLine());

string[] names = new string[EMPLOYEES];
int[,] hoursWorkedPerDay = new int[EMPLOYEES, DAYS];
for (i = 0; i <= EMPLOYEES - 1; i++) {
  names[i] = Console.ReadLine();
  for (j = 0; j <= DAYS - 1; j++) {
    hoursWorkedPerDay[i, j] = Convert.ToInt32(Console.ReadLine());
  }
}

int[] hoursWorkedPerWeek = new int[EMPLOYEES];
for (i = 0; i <= EMPLOYEES - 1; i++) {
  hoursWorkedPerWeek[i] = 0;
  for (j = 0; j <= DAYS - 1; j++) {
    hoursWorkedPerWeek[i] += hoursWorkedPerDay[i, j];
  }
  if (hoursWorkedPerWeek[i] > 40) {
    Console.WriteLine(names[i]);
  }
}

totalGrossPay = 0;
for (i = 0; i <= EMPLOYEES - 1; i++) {
  if (hoursWorkedPerWeek[i] <= 40) {
    grossPay = payRate * hoursWorkedPerWeek[i];
  }
  else {
    grossPay = payRate * 40 + 1.5 * payRate * (hoursWorkedPerWeek[i] - 40);
  }
  totalGrossPay += grossPay;
  Console.WriteLine(names[i] + ", " + (grossPay / 5));
}

Console.WriteLine(totalGrossPay);

for (i = 0; i <= EMPLOYEES - 1; i++) {
  if (hoursWorkedPerWeek[i] > 40) {
    for (j = 0; j <= DAYS - 1; j++) {
      if (hoursWorkedPerDay[i, j] > 8) {
```

```
        Console.WriteLine(names[i] + ", " + days[j] + " Overtime!");
      }
    }
  }
}

for (j = 0; j <= DAYS - 1; j++) {
  total = 0;
  for (i = 0; i <= EMPLOYEES - 1; i++) {
    if (hoursWorkedPerDay[i, j] <= 8) {
      grossPay = payRate * hoursWorkedPerDay[i, j];
    }
    else {
      grossPay = payRate * 8 + 1.5 * payRate * (hoursWorkedPerDay[i, j] - 8);
    }
    total += grossPay;
  }
  Console.WriteLine(days[j] + ", " + total);
}
```

12. Solution

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

int i, j, k;

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

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

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

13. Solution

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

int i, j, k;

int[] a = {16, 12, 3, 5, 6, 9, 18, 19, 20};
```

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

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

Chapter 34

34.7 Review Questions: True/False

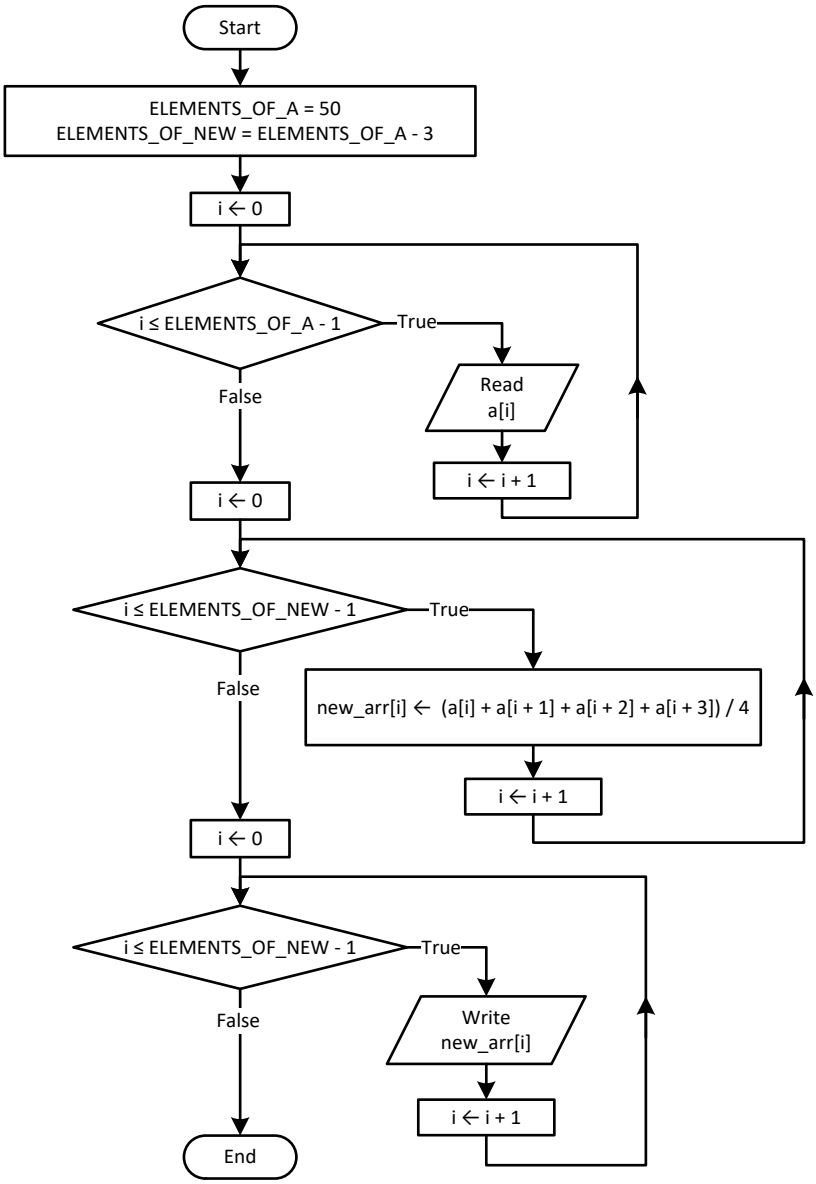
1. true
2. false
3. true
4. true
5. true
6. true
7. true
8. false
9. true
10. false
11. false
12. true
13. false
14. false
15. false
16. true
17. true
18. true
19. false
20. false
21. true
22. true
23. true
24. false
25. true
26. false
27. false
28. false
29. true
30. true
31. true
32. false
33. true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

34.8 Review Exercises

1. Solution

```
for (i = 0; i <= ROWS - 1; i++) {
  for (j = 0; j <= COLUMNS - 1; j++) {
    a[i, j] = Convert.ToDouble(Console.ReadLine());
    while (a[i, j] == 0) {
      Console.WriteLine("Error");
      a[i, j] = Convert.ToDouble(Console.ReadLine());
    }
  }
}
```

2. Solution

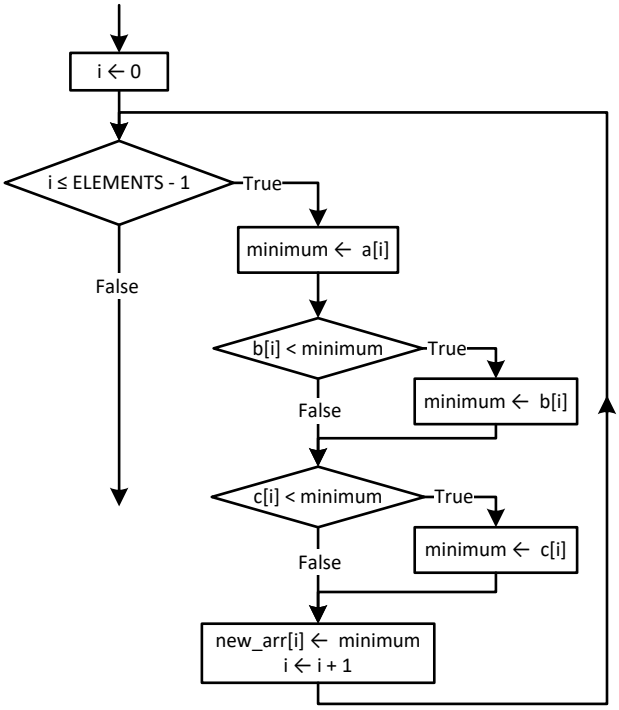



```
const int ELEMENTS_OF_A = 50;
const int ELEMENTS_OF_NEW = ELEMENTS_OF_A - 3;

int i;

double[] a = new double[ELEMENTS_OF_A];
for (i = 0; i <= ELEMENTS_OF_A - 1; i++) {
  a[i] = Convert.ToDouble(Console.ReadLine());
}

double[] newArr = new double[ELEMENTS_OF_NEW];
for (i = 0; i <= ELEMENTS_OF_NEW - 1; i++) {
  newArr[i] = (a[i] + a[i + 1] + a[i + 2] + a[i + 3]) / 4;
}
```

```
for (i = 0; i <= ELEMENTS_OF_NEW - 1; i++) {
  Console.WriteLine(newArr[i] + "\t");
}
```

3. Solution

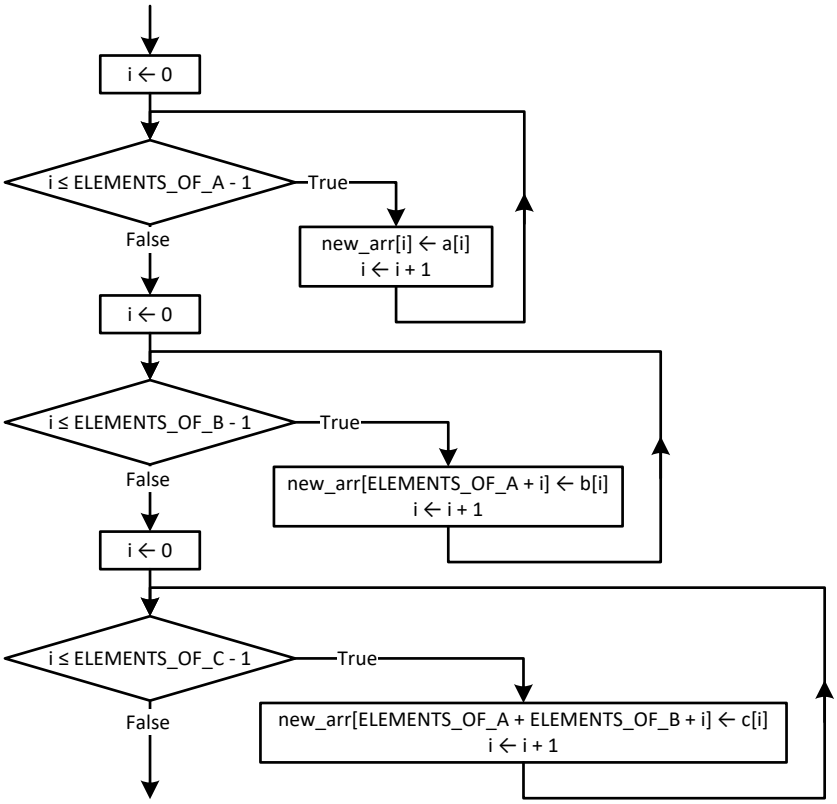
```
const int ELEMENTS = 15;

int i;
double minimum

double[] a = new double[ELEMENTS];
for (i = 0; i <= ELEMENTS - 1; i++) {
  a[i] = Convert.ToDouble(Console.ReadLine());
}
double[] b = new double[ELEMENTS];
for (i = 0; i <= ELEMENTS - 1; i++) {
  b[i] = Convert.ToDouble(Console.ReadLine());
}
double[] c = new double[ELEMENTS];
for (i = 0; i <= ELEMENTS - 1; i++) {
  c[i] = Convert.ToDouble(Console.ReadLine());
}

double[] newArr = new double[ELEMENTS];
for (i = 0; i <= ELEMENTS - 1; i++) {
  minimum = a[i];
  if (b[i] < minimum) {
    minimum = b[i];
  }
  if (c[i] < minimum) {
    minimum = c[i];
  }
  newArr[i] = minimum;
}

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



4. 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;

int i;

double[] a = new double[ELEMENTS_OF_A];
for (i = 0; i <= ELEMENTS_OF_A - 1; i++) {
  a[i] = Convert.ToDouble(Console.ReadLine());
}
double[] b = new double[ELEMENTS_OF_B];
for (i = 0; i <= ELEMENTS_OF_B - 1; i++) {
  b[i] = Convert.ToDouble(Console.ReadLine());
}
double[] c = new double[ELEMENTS_OF_C];
for (i = 0; i <= ELEMENTS_OF_C - 1; i++) {
  c[i] = Convert.ToDouble(Console.ReadLine());
}

double[] newArr = new double[ELEMENTS_OF_NEW];
for (i = 0; i <= ELEMENTS_OF_C - 1; i++) {
  newArr[i] = c[i];
}
for (i = 0; i <= ELEMENTS_OF_B - 1; i++) {
  newArr[ELEMENTS_OF_C + i] = b[i];
}
```

```
for (i = 0; i <= ELEMENTS_OF_A - 1; i++) {
  newArr[ELEMENTS_OF_B + ELEMENTS_OF_C + i] = a[i];
}

//Display array new
for (i = 0; i <= ELEMENTS_OF_NEW - 1; i++) {
  Console.Write(newArr[i] + "\t");
}
```



5. Solution

```
const int COLUMNS = 4;

int i, j;

//Create arrays a and b
int[,] a = {
  {10, 11, 12, 85},
  {3, 1, 5, 10},
  {-1, 2, -5, -10}
};
int[,] b = {
  {10, 11, 16, 33},
  {11, 13, 5, 55},
  {-1, -2, -4, 44},
  {55, 33, 77, 12},
  {-110, 120, 132, 43}
```

```
};

int rowsOfA = a.Length / COLUMNS;
int rowsOfB = b.Length / COLUMNS;

//Create array newArr
int[,] newArr = new int[rowsOfA + rowsOfB, COLUMNS];
for (i = 0; i <= rowsOfA - 1; i++) {
  for (j = 0; j <= COLUMNS - 1; j++) {
    newArr[i, j] = a[i, j];
  }
}
for (i = 0; i <= rowsOfB - 1; i++) {
  for (j = 0; j <= COLUMNS - 1; j++) {
    newArr[rowsOfA + i, j] = b[i, j];
  }
}

//Display array newArr
for (i = 0; i <= rowsOfA + rowsOfB - 1; i++) {
  for (j = 0; j <= COLUMNS - 1; j++) {
    Console.Write(newArr[i, j] + "\t");
  }
  Console.WriteLine();
}
```

6. Solution

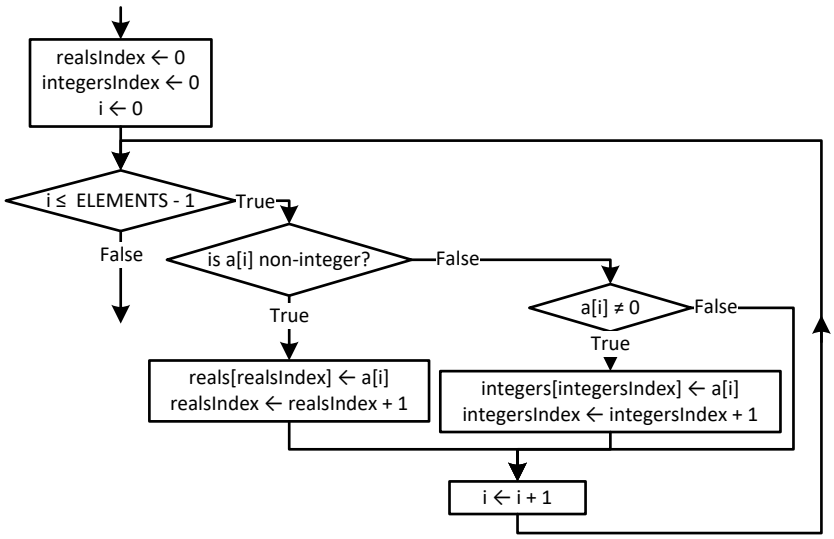
```
const int COLUMNS_OF_A = 10;
const int COLUMNS_OF_B = 15;
const int COLUMNS_OF_C = 20;
const int ROWS = 5;
const int COLUMNS = COLUMNS_OF_A + COLUMNS_OF_B + COLUMNS_OF_C;

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] = Convert.ToDouble(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] = Convert.ToDouble(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] = Convert.ToDouble(Console.ReadLine());
  }
}

double[,] newArr = new double[ROWS, COLUMNS];
for (i = 0; i <= ROWS - 1; i++) {
  for (j = 0; j <= COLUMNS_OF_A - 1; j++) {
    newArr[i, j] = a[i, j];
  }
}
for (i = 0; i <= ROWS - 1; i++) {
  for (j = 0; j <= COLUMNS_OF_B - 1; j++) {
    newArr[i, COLUMNS_OF_A + j] = b[i, j];
  }
}
for (i = 0; i <= ROWS - 1; i++) {
  for (j = 0; j <= COLUMNS_OF_C - 1; j++) {
    newArr[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(newArr[i, j] + "\t");
  }
  Console.WriteLine();
}
```

7. Solution

```
const int ELEMENTS = 50;

int i, integersIndex, realsIndex;

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

double[] reals = new double[ELEMENTS];
int[] integers = new int[ELEMENTS];
realsIndex = 0;
integersIndex = 0;
for (i = 0; i <= ELEMENTS - 1; i++) {
  if (a[i] != (int)(a[i])) {
    reals[realsIndex] = a[i];
    realsIndex++;
  }
  else if (a[i] != 0) {
    integers[integersIndex] = (int)a[i];
    integersIndex++;
  }
```

```
}

for (i = 0; i <= realsIndex - 1; i++) {
  Console.Write(reals[i] + "\t");
}

Console.WriteLine();
for (i = 0; i <= integersIndex - 1; i++) {
  Console.Write(integers[i] + "\t");
}
```



8. Solution

```
const int ELEMENTS = 50;

int digit1, digit2, digit3, i, k, r;

int[] a = new int[ELEMENTS];
for (i = 0; i <= ELEMENTS - 1; i++) {
  a[i] = Convert.ToInt32(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");
}
```

9. Solution

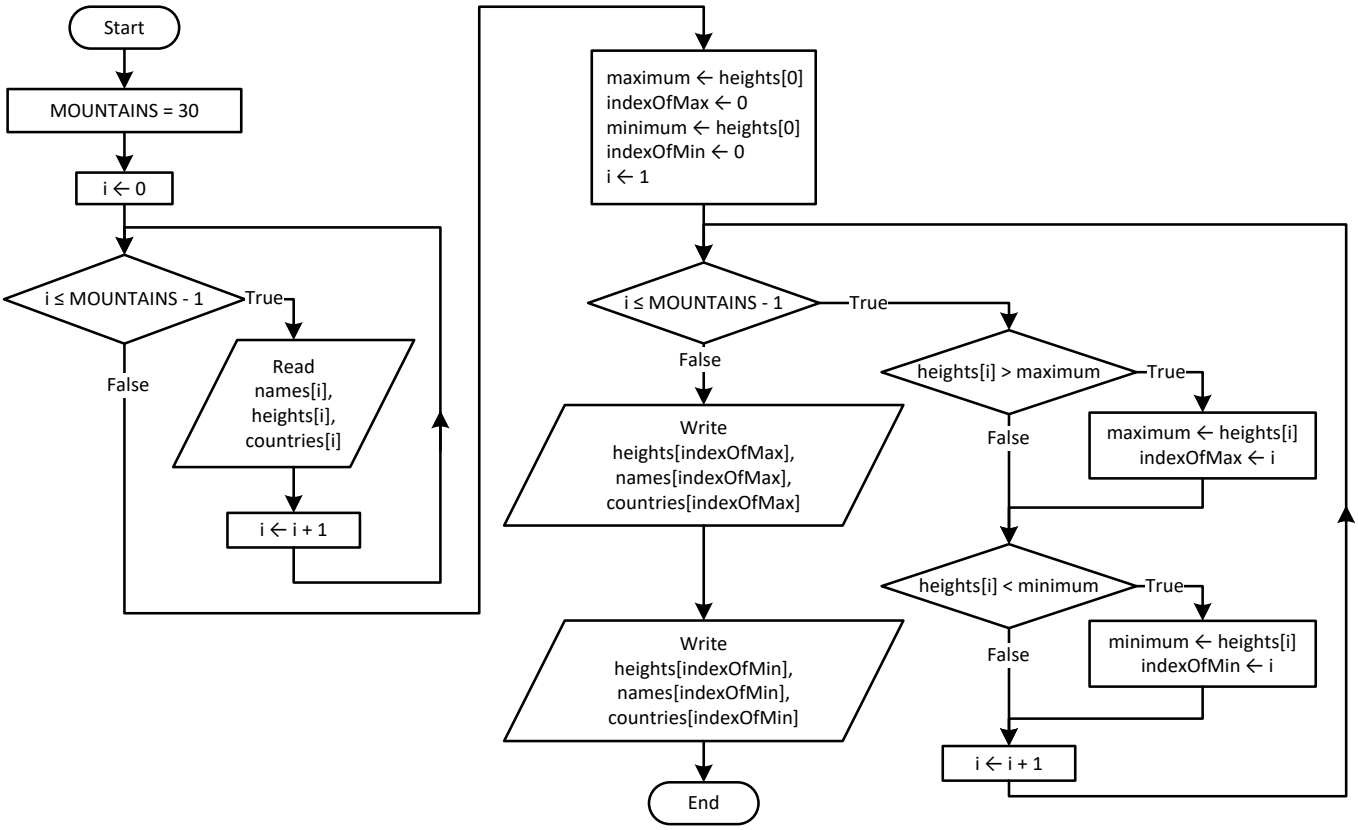
```
const int PRODUCTS = 10;
const int CITIZENS = 200;

int countB, i, j, maximum;

string[] prodNames = new string[PRODUCTS];
string[,] answers = new string[PRODUCTS, CITIZENS];
for (i = 0; i <= PRODUCTS - 1; i++) {
  prodNames[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[] countA = new int[PRODUCTS];
for (i = 0; i <= PRODUCTS - 1; i++) {
  countA[i] = 0;
  for (j = 0; j <= CITIZENS - 1; j++) {
    if (answers[i, j] == "A") {
      countA[i]++;
    }
  }
  Console.WriteLine(prodNames[i] + ", " + countA[i]);
}

for (j = 0; j <= CITIZENS - 1; j++) {
  countB = 0;
  for (i = 0; i <= PRODUCTS - 1; i++) {
    if (answers[i, j] == "B") {
      countB++;
    }
  }
  Console.WriteLine(countB);
}

maximum = countA[0];
for (i = 1; i <= PRODUCTS - 1; i++) {
  if (countA[i] > maximum) {
    maximum = countA[i];
  }
}
```

```
for (i = 0; i <= PRODUCTS - 1; i++) {
  if (countA[i] == maximum) {
    Console.WriteLine(prodNames[i]);
  }
}
```

10. Solution

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

int i, j, minJ;
double minimum;

string[] usNames = new string[US_CITIES];
for (i = 0; i <= US_CITIES - 1; i++) {
  Console.Write("Enter name for US city No " + (i + 1) + ": ");
  usNames[i] = Console.ReadLine();
}

string[] canadianNames = new string[CANADIAN_CITIES];
for (j = 0; j <= CANADIAN_CITIES - 1; j++) {
  Console.Write("Enter name for Canadian city No " + (j + 1) + ": ");
  canadianNames[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.Write("Enter distance between " + usNames[i] + " and " + canadianNames[j] + ": ");
    distances[i, j] = Convert.ToDouble(Console.ReadLine());
  }
}

for (i = 0; i <= US_CITIES - 1; i++) {
  minimum = distances[i, 0];
  minJ = 0;
  for (j = 1; j <= CANADIAN_CITIES - 1; j++) {
    if (distances[i, j] < minimum) {
      minimum = distances[i, j];
      minJ = j;
    }
  }
  Console.WriteLine("Closest Canadian city to " + usNames[i] + " is " + canadianNames[minJ]);
}
```

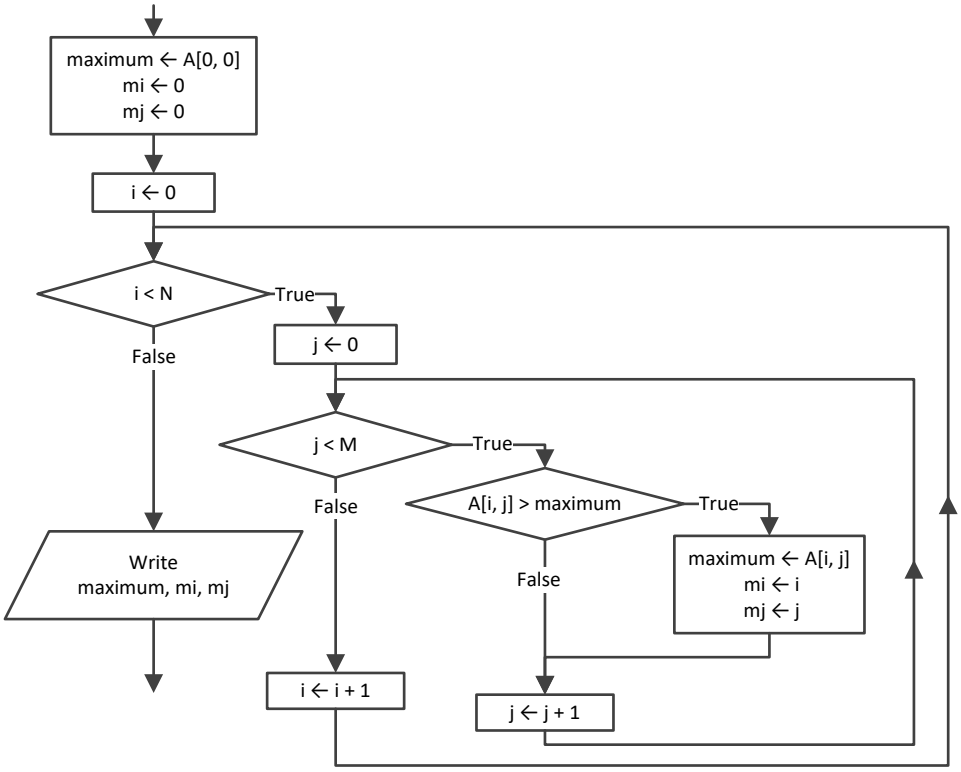
11. Solution




```
const int MOUNTAINS = 30;

int i, indexOfMax, indexOfMin;
double maximum, minimum;

string[] names = new string[MOUNTAINS];
double[] heights = new double[MOUNTAINS];
string[] countries = new string[MOUNTAINS];
for (i = 0; i <= MOUNTAINS - 1; i++) {
  names[i] = Console.ReadLine();
  heights[i] = Convert.ToDouble(Console.ReadLine());
  countries[i] = Console.ReadLine();
}

maximum = heights[0];
indexOfMax = 0;
minimum = heights[0];
indexOfMin = 0;
for (i = 1; i <= MOUNTAINS - 1; i++) {
  if (heights[i] > maximum) {
    maximum = heights[i];
    indexOfMax = i;
  }
  if (heights[i] < minimum) {
    minimum = heights[i];
```

```
    indexOfMin = i;
  }
}

Console.WriteLine(heights[indexOfMax] + ", " + names[indexOfMax] + ", " + countries[indexOfMax]);
Console.WriteLine(heights[indexOfMin] + ", " + names[indexOfMin] + ", " + countries[indexOfMin]);
```

12. Solution



13. Solution

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

int i, j, m_i, maximum;

string[] names = new string[TEAMS];
string[,] results = new string[TEAMS, GAMES];
for (i = 0; i <= TEAMS - 1; i++) {
  names[i] = Console.ReadLine();
  for (j = 0; j <= GAMES - 1; j++) {
    results[i, j] = Console.ReadLine();
  }
}

int[] points = new int[TEAMS];
for (i = 0; i <= TEAMS - 1; i++) {
  points[i] = 0;
```

```
  for (j = 0; j <= GAMES - 1; j++) {
    if (results[i, j] == "W") {
      points[i] += 3;
    }
    else if (results[i, j] == "T") {
      points[i] += 1;
    }
  }
}

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

Console.WriteLine(names[m_i]);
```

14. Solution

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

int i, j;
double maxi, mini;

double[,] heights = new double[OBJECTS, FALLS];
double[,] times = new double[OBJECTS, FALLS];
for (i = 0; i <= OBJECTS - 1; i++) {
  for (j = 0; j <= FALLS - 1; j++) {
    heights[i, j] = Convert.ToDouble(Console.ReadLine());
    times[i, j] = Convert.ToDouble(Console.ReadLine());
  }
}

double[,] g = new double[OBJECTS, FALLS];
for (i = 0; i <= OBJECTS - 1; i++) {
  for (j = 0; j <= FALLS - 1; j++) {
    g[i, j] = 2 * heights[i, j] / Math.Pow(times[i, j], 2);
  }
}

double[] minimum = new double[OBJECTS];
double[] maximum = new double[OBJECTS];
for (i = 0; i <= OBJECTS - 1; i++) {
  minimum[i] = g[i, 0];
  maximum[i] = g[i, 0];
  for (j = 1; j <= FALLS - 1; j++) {
    if (g[i, j] < minimum[i]) {
      minimum[i] = g[i, j];
```

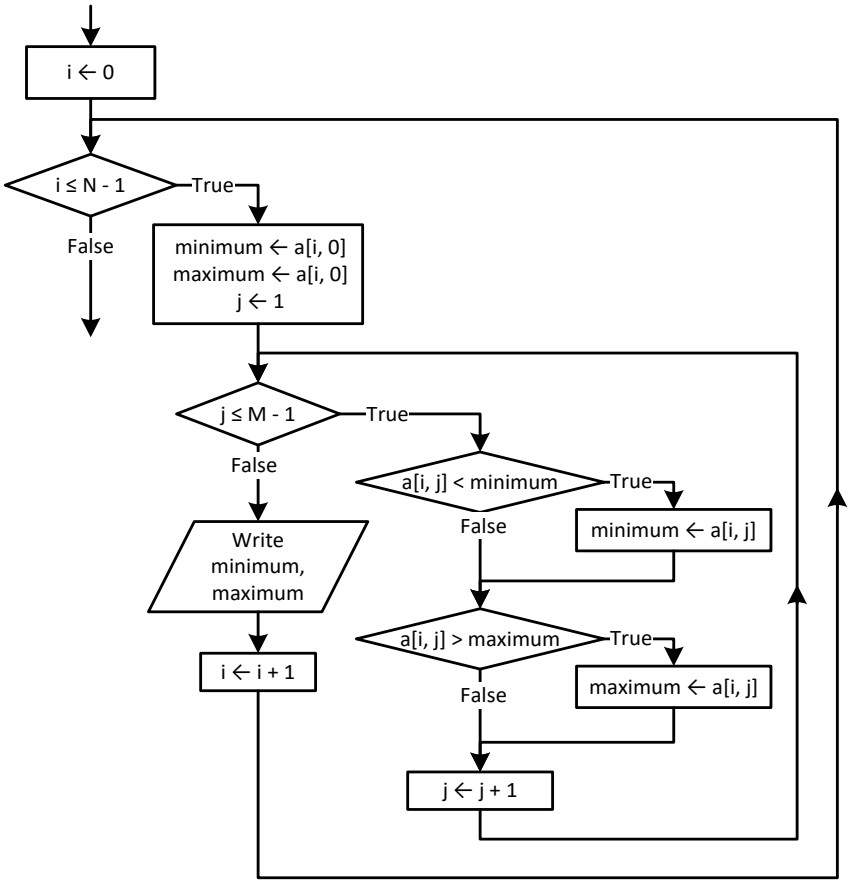
```
    }
    if (g[i, j] > maximum[i]) {
      maximum[i] = g[i, j];
    }
  }
}

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

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

Console.WriteLine(mini + ", " + maxi);
```

15. Solution

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

int i, j, m_i;
double minimum;

string[] names = new string[STATIONS];
double[,] co2 = new double[STATIONS, DAYS];
for (i = 0; i <= STATIONS - 1; i++) {
  names[i] = Console.ReadLine();
  for (j = 0; j <= DAYS - 1; j++) {
    co2[i, j] = Convert.ToDouble(Console.ReadLine());
  }
}

double[] average = new double[STATIONS];
for (i = 0; i <= STATIONS - 1; i++) {
  average[i] = 0;
  for (j = 0; j <= DAYS - 1; j++) {
    average[i] += co2[i, j];
  }
  average[i] /= DAYS;
}

minimum = average[0];
m_i = 0;
for (i = 1; i <= STATIONS - 1; i++) {
```

```
  if (average[i] < minimum) {
    minimum = average[i];
    m_i = i;
  }
}
Console.WriteLine(names[m_i]);
```

16. Solution



17. Solution

First approach

```
const int ROWS = 20;
const int COLUMNS = 30;

int i, j;

double[,] b = new double[ROWS, COLUMNS];
for (i = 0; i <= ROWS - 1; i++) {
  for (j = 0; j <= COLUMNS - 1; j++) {
    b[i, j] = Convert.ToDouble(Console.ReadLine());
  }
}

double[] minimum = new double[COLUMNS];
```

```
double[] maximum = new double[COLUMNS];
for (j = 0; j <= COLUMNS - 1; j++) {
  minimum[j] = b[0, j];
  maximum[j] = b[0, j];
  for (i = 1; i <= ROWS - 1; i++) {
    if (b[i, j] < minimum[j]) {
      minimum[j] = b[i, j];
    }
    if (b[i, j] > maximum[j]) {
      maximum[j] = b[i, j];
    }
  }
}

for (j = 0; j <= COLUMNS - 1; j++) {
  Console.WriteLine(minimum[j] + " " + maximum[j]);
}
```

Second approach

```
const int ROWS = 20;
const int COLUMNS = 30;

int i, j;
double minimum, maximum;

double[,] b = new double[ROWS, COLUMNS];
for (i = 0; i <= ROWS - 1; i++) {
  for (j = 0; j <= COLUMNS - 1; j++) {
    b[i, j] = Convert.ToDouble(Console.ReadLine());
  }
}

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

18. Solution

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

int i, j, m, n, temp;
bool swaps;
```

```
string tempStr;

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.Write("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 <= 3; m++) {   //Perfom only three passes
  swaps = false;
  for (n = TEAMS - 1; n >= m; n--) {
    if (points[n] > points[n - 1]) {
      temp = points[n];
      points[n] = points[n - 1];
      points[n - 1] = temp;

      tempStr = names[n];
      names[n] = names[n - 1];
      names[n - 1] = tempStr;

      swaps = true;
    }
  }
  if (!swaps) break;
}

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

19. Solution

```
const int PEOPLE = 50;

int i, m, n;
double temp;
string tempStr;

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] = Convert.ToDouble(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;

      tempStr = names[n];
      names[n] = names[n - 1];
      names[n - 1] = tempStr;
    }
    else if (heights[n] == heights[n - 1]) {
      if (names[n].CompareTo(names[n - 1]) < 0) {
        tempStr = names[n];
        names[n] = names[n - 1];
        names[n - 1] = tempStr;
      }
    }
  }
}

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

20. Solution

```
const int PEOPLE = 50;

string tempStr;
int i, m, n;

string[] firstNames = new string[PEOPLE];
string[] lastNames = new string[PEOPLE];
string[] fatherNames = new string[PEOPLE];
```

```
for (i = 0; i <= PEOPLE - 1; i++) {
  Console.Write("Enter first name for person No." + (i + 1) + ": ");
  firstNames[i] = Console.ReadLine();
  Console.Write("Enter last name for person No." + (i + 1) + ": ");
  lastNames[i] = Console.ReadLine();
  Console.Write("Enter father's name for person No." + (i + 1) + ": ");
  fatherNames[i] = Console.ReadLine();
}

for (m = 1; m <= PEOPLE - 1; m++) {
  for (n = PEOPLE - 1; n >= m; n--) {
    if (lastNames[n].CompareTo(lastNames[n - 1]) < 0) {
      tempStr = lastNames[n];
      lastNames[n] = lastNames[n - 1];
      lastNames[n - 1] = tempStr;

      tempStr = firstNames[n];
      firstNames[n] = firstNames[n - 1];
      firstNames[n - 1] = tempStr;

      tempStr = fatherNames[n];
      fatherNames[n] = fatherNames[n - 1];
      fatherNames[n - 1] = tempStr;
    }
    else if (lastNames[n] == lastNames[n - 1]) {
      if (firstNames[n].CompareTo(firstNames[n - 1]) < 0) {
        tempStr = firstNames[n];
        firstNames[n] = firstNames[n - 1];
        firstNames[n - 1] = tempStr;

        tempStr = fatherNames[n];
        fatherNames[n] = fatherNames[n - 1];
        fatherNames[n - 1] = tempStr;
      }
      else if (firstNames[n] == firstNames[n - 1]) {
        if (fatherNames[n].CompareTo(fatherNames[n - 1]) < 0) {
          tempStr = fatherNames[n];
          fatherNames[n] = fatherNames[n - 1];
          fatherNames[n - 1] = tempStr;
        }
      }
    }
  }
}

for (i = 0; i <= PEOPLE - 1; i++) {
  Console.WriteLine(lastNames[i] + "\t" + firstNames[i] + "\t" + fatherNames[i]);
}
```

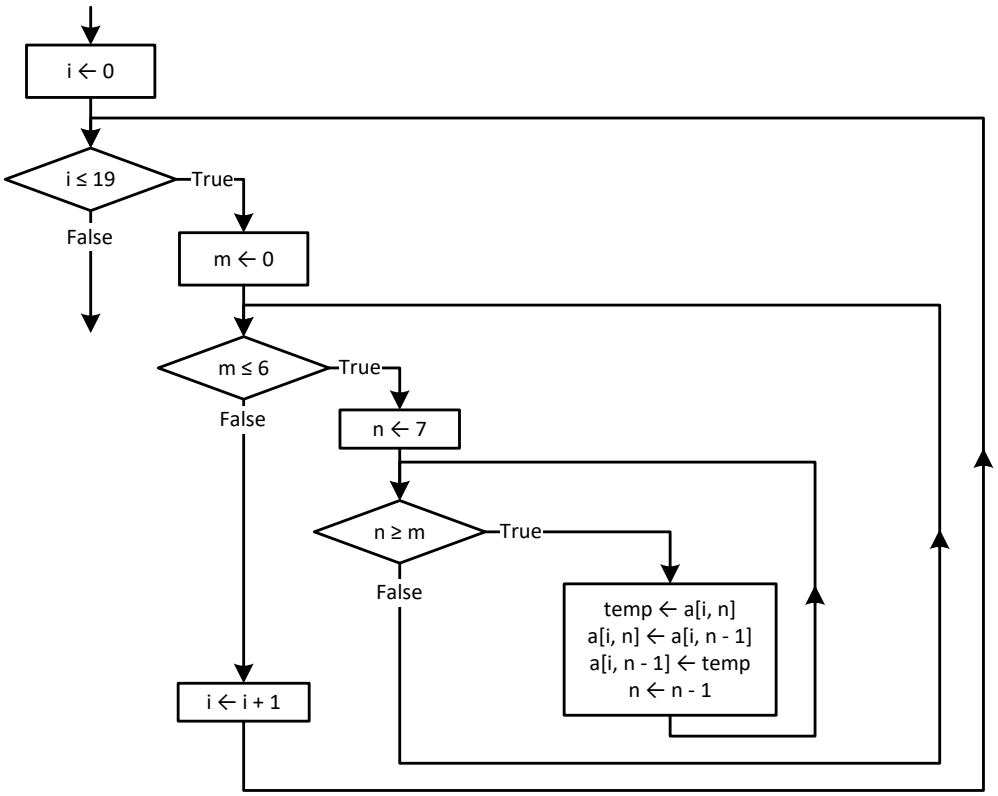
21. Solution

```
const int ARTISTS = 12;
```

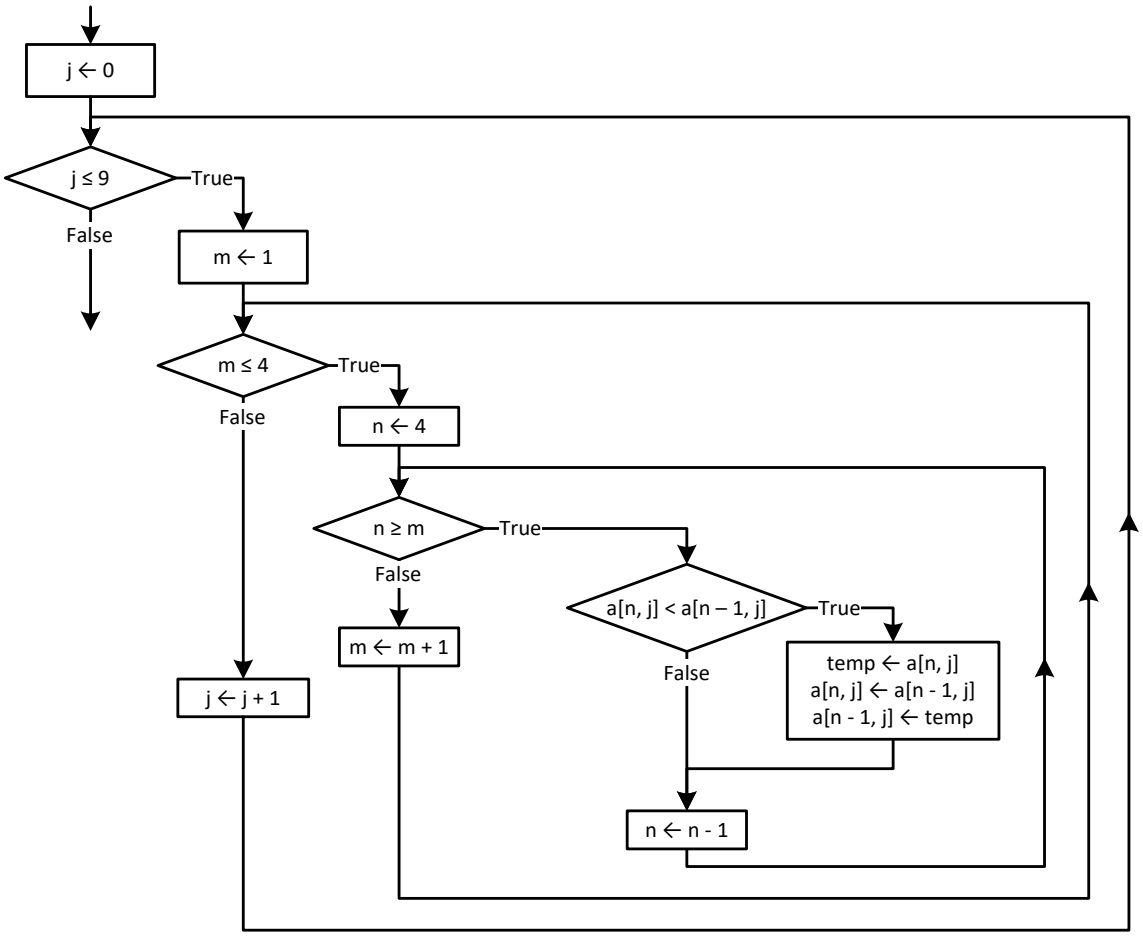
```
const int JUDGES = 10;

int i, j, m, maximum, minimum, n, temp;
string tempStr;

string[] artistNames = 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) + ": ");
  artistNames[i] = Console.ReadLine();
  for (j = 0; j <= JUDGES - 1; j++) {
    Console.Write("Enter score for artist: " + artistNames[i]);
    Console.Write(" gotten from judge No " + (j + 1) + ": ");
    score[i, j] = Convert.ToInt32(Console.ReadLine());
  }
}

int[] total = new int[ARTISTS];
for (i = 0; i <= ARTISTS - 1; i++) {
  total[i] = 0;
  for (j = 1; j <= JUDGES - 1; j++) {
    total[i] += score[i, j];
  }
}

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

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

      tempStr = artistNames[n];
      artistNames[n] = artistNames[n - 1];
      artistNames[n - 1] = tempStr;
    }
    else if (total[n] == total[n - 1]) {
```

```
      if (artistNames[n].CompareTo(artistNames[n - 1]) < 0) {
        tempStr = artistNames[n];
        artistNames[n] = artistNames[n - 1];
        artistNames[n - 1] = tempStr;
      }
    }
  }
}

for (i = 0; i <= ARTISTS - 1; i++) {
  Console.WriteLine(artistNames[i] + ", " + total[i]);
}
```

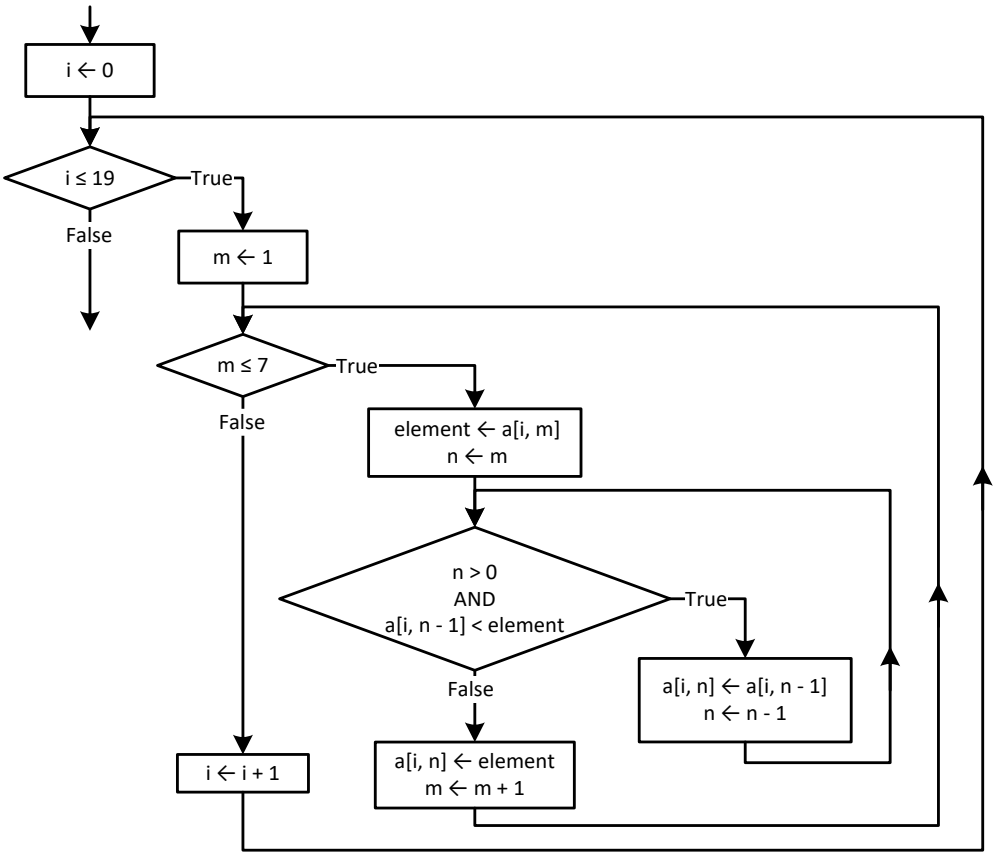
22. Solution



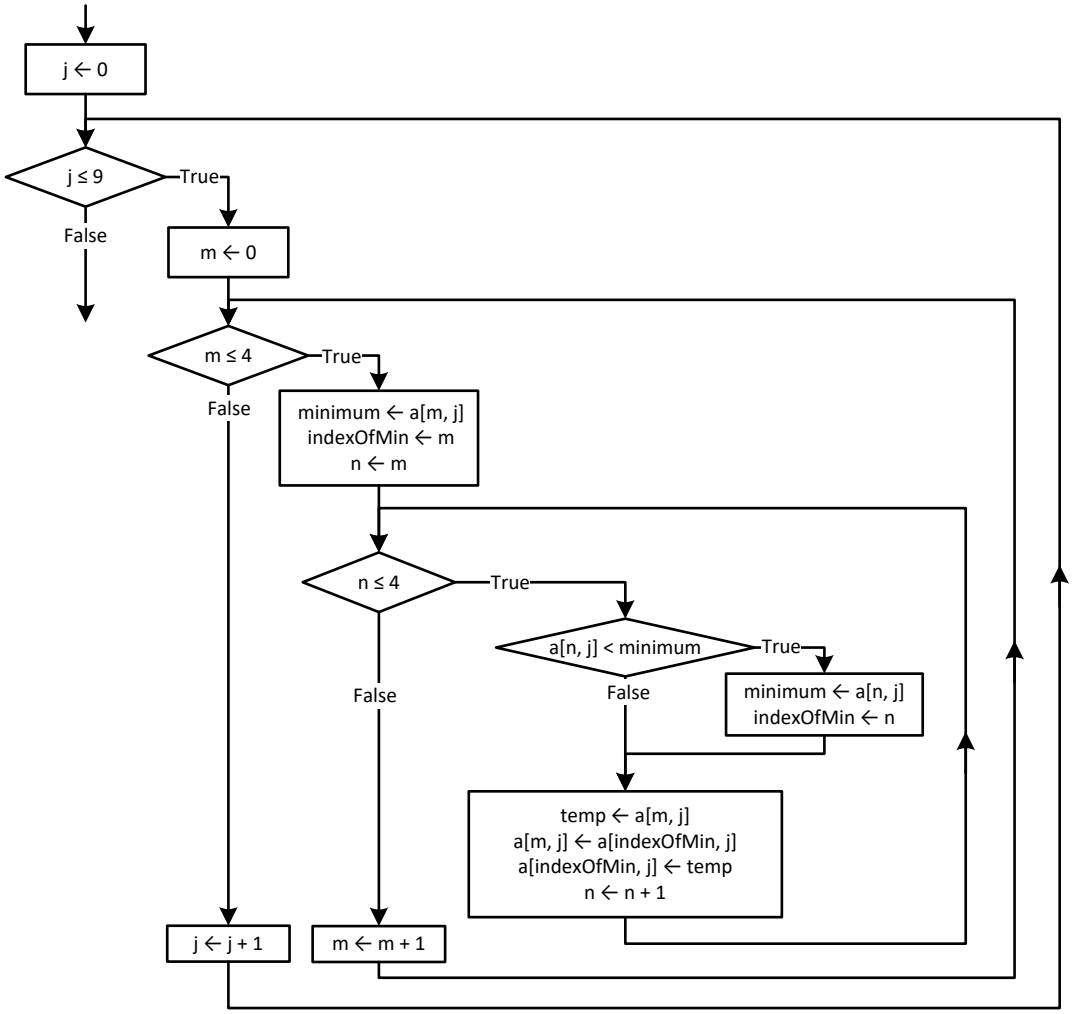
23. Solution


j ← 0
j ≤ 9
True
False
m ← 1
m ≤ 4
True
False
n ← 4
n ≥ m
True
False
a[n, j] < a[n – 1, j]
True
False
temp ← a[n, j]
a[n, j] ← a[n - 1, j]
a[n - 1, j] ← temp
m ← m + 1
j ← j + 1
n ← n - 1

24. Solution


i ← 0
i ≤ 19
True
False
m ← 1
m ≤ 7
True
False
element ← a[i, m]
n ← m
n > 0
AND
a[i, n - 1] < element
True
False
a[i, n] ← a[i, n - 1]
n ← n - 1
a[i, n] ← element
m ← m + 1
i ← i + 1

25. Solution



26. Solution

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

int i, indexOfMin, j, m, n, hours, minutes, seconds, iTemp;
double minimum, dTemp;
string tempStr;

string[] names = new string[PEOPLE];
int[,] times = new int[PEOPLE, PUZZLES];
for (i = 0; i <= PEOPLE - 1; i++) {
  names[i] = Console.ReadLine();
  for (j = 0; j <= PUZZLES - 1; j++) {
    hours = Convert.ToInt32(Console.ReadLine());
    minutes = Convert.ToInt32(Console.ReadLine());
    seconds = Convert.ToInt32(Console.ReadLine());
    times[i, j] = hours * 3600 + minutes * 60 + seconds;
```

```
  }
}

for (i = 0; i <= PEOPLE - 1; i++) {
  for (m = 0; m <= PUZZLES - 1; m++) {
    minimum = times[i, m];
    indexOfMin = m;
    for (n = m; n <= PUZZLES - 1; n++) {
      if (times[i, n] < minimum) {
        minimum = times[i, n];
        indexOfMin = n;
      }
    }
    iTemp = times[i, m];
    times[i, m] = times[i, indexOfMin];
    times[i, indexOfMin] = iTemp;
  }
}

for (i = 0; i <= PEOPLE - 1; i++) {
  Console.WriteLine(names[i]);
  for (j = 0; j <= 2; j++) {
    Console.WriteLine(times[i, j]);
  }
}

double[] average = new double[PEOPLE];
for (i = 0; i <= PEOPLE - 1; i++) {
  average[i] = 0;
  for (j = 0; j <= PUZZLES - 1; j++) {
    average[i] += times[i, j];
  }
  average[i] /= PUZZLES;
}

for (m = 0; m <= 2; m++) {   //Perform only 3 iterations
  minimum = average[m];
  indexOfMin = m;
  for (n = m; n <= PEOPLE - 1; n++) {
    if (average[n] < minimum) {
      minimum = average[n];
      indexOfMin = n;
    }
  }
  dTemp = average[m];
  average[m] = average[indexOfMin];
  average[indexOfMin] = dTemp;

  tempStr = names[m];
  names[m] = names[indexOfMin];
  names[indexOfMin] = tempStr;
}
```

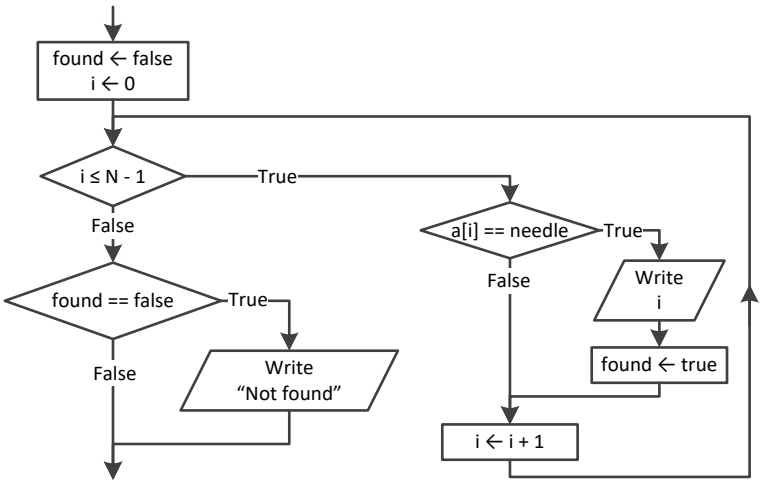
```
Console.WriteLine(names[0] + ", " + names[1] + ", " + names[2]);
```

27. Solution

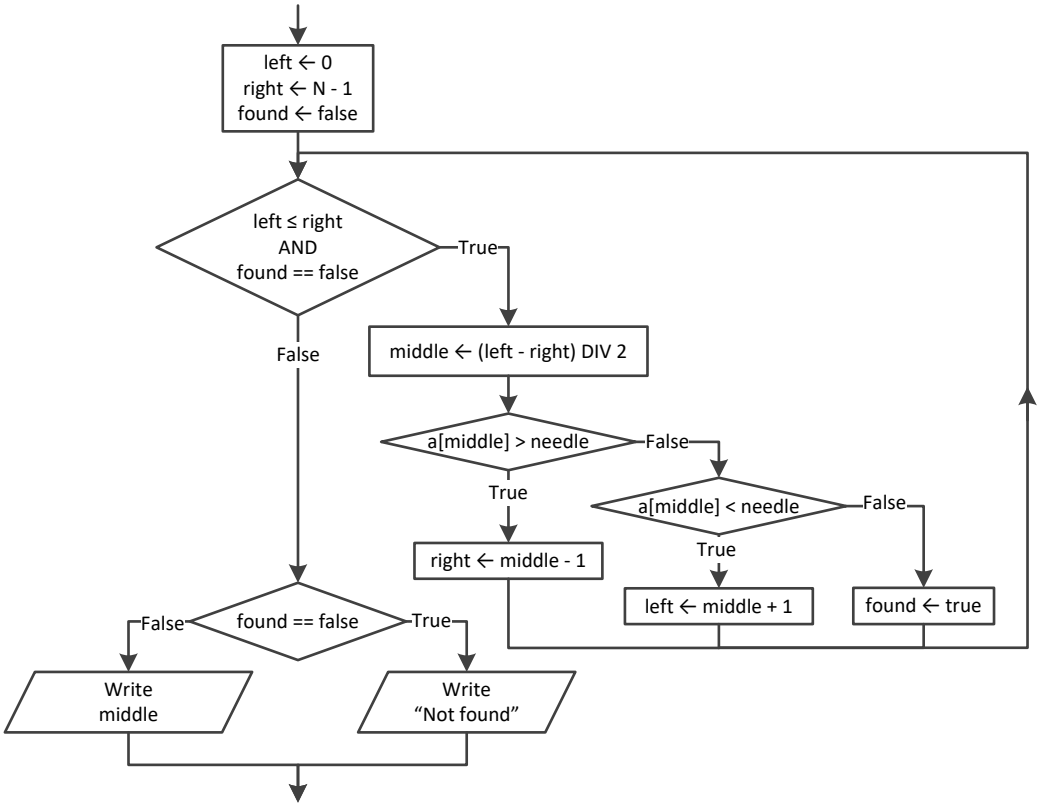
```
const int AREAS = 5;
const int HOURS = 48;

int i, j, m, m_i, m_j, n;
double maximum, element1;
string element2;

string[] names = new string[AREAS];
double[,] CO2 = new double[AREAS, HOURS];
for (i = 0; i <= AREAS - 1; i++) {
  names[i] = Console.ReadLine();
  for (j = 0; j <= HOURS - 1; j++) {
    CO2[i, j] = Convert.ToDouble(Console.ReadLine());
  }
}

double[] averagePerHour = new double[AREAS];
for (i = 0; i <= AREAS - 1; i++) {
  averagePerHour[i] = 0;
  for (j = 0; j <= HOURS - 1; j++) {
    averagePerHour[i] += CO2[i, j];
  }
  averagePerHour[i] /= HOURS;
}

for (i = 0; i <= AREAS - 1; i++) {
  Console.WriteLine(names[i] + ", " + averagePerHour[i]);
}

double[] averagePerCity = new double[HOURS];
for (j = 0; j <= HOURS - 1; j++) {
  averagePerCity[j] = 0;
  for (i = 0; i <= AREAS - 1; i++) {
    averagePerCity[j] += CO2[i, j];
  }
  averagePerCity[j] /= AREAS;
}

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

maximum = averagePerCity[0];
m_j = 0;
for (j = 1; j <= HOURS - 1; j++) {
  if (averagePerCity[j] > maximum) {
    maximum = averagePerCity[j];
    m_j = j;
```

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

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

for (m = 1; m <= AREAS - 1; m++) {
  element1 = averagePerHour[m];
  element2 = names[m];

  n = m;
  while (n > 0 && averagePerHour[n - 1] < element1) {
    averagePerHour[n] = averagePerHour[n - 1];
    names[n] = names[n - 1];
    n--;
  }

  averagePerHour[n] = element1;
  names[n] = element2;
}

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

28. Solution



29. Solution



30. Solution

```
const int TEAMS = 20;
const int WEEKS = 12;

int i, j;
string needle;
bool found;

string[] names = new string[TEAMS];
string[,] results = new string[TEAMS, WEEKS];
for (i = 0; i <= TEAMS - 1; i++) {
  Console.Write("Enter name for team No. " + (i + 1) + ": ");
  names[i] = Console.ReadLine();
  for (j = 0; j <= WEEKS - 1; j++) {
    Console.Write("Enter result for");
    Console.Write(" week No. " + (j + 1) + " for " + names[i] + ": ");
    results[i, j] = Console.ReadLine();
  }
}

//Get value to search and convert it to uppercase
Console.Write("Enter a result to search: ");
needle = Console.ReadLine().ToUpper();
```

```
for (i = 0; i <= TEAMS - 1; i++) {
  found = false;
  Console.WriteLine("Found results for " + names[i]);
  for (j = 0; j <= WEEKS - 1; j++) {
    if (results[i, j].ToUpper() == needle) {
      Console.WriteLine("Week " + (j + 1));
      found = true;
    }
  }

  if (!found) {
    Console.WriteLine("No results!");
  }
}
```

31. Solution

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

int i, j, total;
string needle;

string[] names = new string[TEAMS];
int[,] goalsScored = new int[TEAMS, GAMES];
int[,] goalsLetIn = 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: ");
    goalsScored[i, j] = Convert.ToInt32(Console.ReadLine());
    while (goalsScored[i, j] < 0) {
      Console.Write("Error! Enter goals scored: ");
      goalsScored[i, j] = Convert.ToInt32(Console.ReadLine());
    }

    Console.Write("Enter goals let in: ");
    goalsLetIn[i, j] = Convert.ToInt32(Console.ReadLine());
    while (goalsLetIn[i, j] < 0) {
      Console.Write("Error! Enter goals let in: ");
      goalsLetIn[i, j] = Convert.ToInt32(Console.ReadLine());
    }
  }
}

Console.Write("Enter a team to search: ");
needle = Console.ReadLine();

i = 0;
while (i < TEAMS - 1 && names[i] != needle) {
  i++;
}
```

```
if (names[i] != needle) {
  Console.WriteLine("This team does not exist");
}
else {
  total = 0;
  for (j = 0; j <= GAMES - 1; j++) {
    if (goalsScored[i, j] > goalsLetIn[i, j]) {
      total += 3;
    }
    else if (goalsScored[i, j] == goalsLetIn[i, j]) {
      total += 1;
    }
  }
  Console.WriteLine(total);
}
```

32. Solution

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

int i, left, m, middle, n, right;
string temp, needle;
bool found;

Console.WriteLine("Class 1");
string[] names1 = new string[CLASS1];
for (i = 0; i <= CLASS1 - 1; i++) {
  Console.Write("Enter name: ");
  names1[i] = Console.ReadLine();
}
Console.WriteLine("Class 2");
string[] names2 = new string[CLASS2];
for (i = 0; i <= CLASS2 - 1; i++) {
  Console.Write("Enter name: ");
  names2[i] = Console.ReadLine();
}

//Bubble sort
for (m = 1; m <= CLASS1 - 1; m++) {
  for (n = CLASS1 - 1; n >= m; n--) {
    if (names1[n].CompareTo(names1[n - 1]) < 0) {
      temp = names1[n];
      names1[n] = names1[n - 1];
      names1[n] = temp;
    }
  }
}
for (m = 1; m <= CLASS2 - 1; m++) {
  for (n = CLASS2 - 1; n >= m; n--) {
    if (names2[n].CompareTo(names2[n - 1]) < 0) {
```

```
      temp = names2[n];
      names2[n] = names2[n - 1];
      names2[n] = temp;
    }
  }
}

Console.WriteLine("\nClass 1");
for (i = 0; i <= CLASS1 - 1; i++) {
  Console.WriteLine(names1[i]);
}
Console.WriteLine("\nClass 2");
for (i = 0; i <= CLASS2 - 1; i++) {
  Console.WriteLine(names2[i]);
}

Console.Write("Enter a name to search: ");
needle = Console.ReadLine();

left = 0;
right = CLASS1 - 1;
found = false;
while (left <= right && !found) {
  middle = (int)((left + right) / 2);

  if (needle.CompareTo(names1[middle]) < 0) {
    right = middle - 1;
  }
  else if (needle.CompareTo(names1[middle]) > 0) {
    left = middle + 1;
  }
  else {
    found = true;
  }
}

if (found) {
  Console.WriteLine("Student found in Class No 1");
}
else {
  left = 0;
  right = CLASS2 - 1;
  while (left <= right && !found) {
    middle = (int)((left + right) / 2);

    if (needle.CompareTo(names2[middle]) < 0) {
      right = middle - 1;
    }
    else if (needle.CompareTo(names2[middle]) > 0) {
      left = middle + 1;
    }
    else {
```

```
      found = true;
    }
  }

  if (found) {
    Console.WriteLine("Student found in Class No 2");
  }
  else {
    Console.WriteLine("Student not found in either class");
  }
}
```

33. Solution

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

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

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

34. Solution

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

found = false;

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

if (!found) {
  value = Convert.ToInt32(valueStr);
  i = 0;
  while (i < 999 && SSNs[i] != value) {
    i++;
  }

  if (SSNs[i] == value) {
```

```
    found = true;
    Console.WriteLine(names[i]);
  }
}

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

35. Solution

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

int i, j;
bool found, failure;

int[,] grades = new int[STUDENTS, LESSONS];
for (i = 0; i <= STUDENTS - 1; i++) {
  for (j = 0; j <= LESSONS - 1; j++) {
    do {
      grades[i, j] = Convert.ToInt32(Console.ReadLine());
      failure = false;
      if (grades[i, j] < 0) {
        Console.WriteLine("Error! You entered a negative value");
        failure = true;
      }
      else if (grades[i, j] > 100) {
        Console.WriteLine("Error! You entered a value grater than 100");
        failure = true;
      }
    } while (failure);
  }
}

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

found = false;
for (i = 0; i <= STUDENTS - 1; i++) {
  if (average[i] < 70) {
    found = true;
    break;
  }
}

if (found) {
```

```
  Console.WriteLine("There is at least one student that has an average value below 70");
}
```

36. Solution

```
string word, letter;
int i;

Dictionary<string, string> morseAlphabet = new() {
  {"A", ".-"},
  {"B", "-..."},
  {"C", "-.-."},
  {"D", "-.."},
  {"E", "."},
  {"F", "..-."},
  {"G", "--."},
  {"H", "...."},
  {"I", ".."},
  {"J", ".---"},
  {"K", "-.-"},
  {"L", ".-.."},
  {"M", "--"},
  {"N", "-."},
  {"O", "---"},
  {"P", ".--."},
  {"Q", "--.-"},
  {"R", ".-."},
  {"S", "..."},
  {"T", "-"},
  {"U", "..-"},
  {"V", "...-"},
  {"W", ".--"},
  {"X", "-..-"},
  {"Y", "-.--"},
  {"Z", "--.."},
  {" ", "/"}
};

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

for (i = 0; i <= word.Length - 1; i++) {
  letter = "" + word[i];
  Console.Write(morseAlphabet[letter.ToUpper()] + " ");
}
```

37. Solution

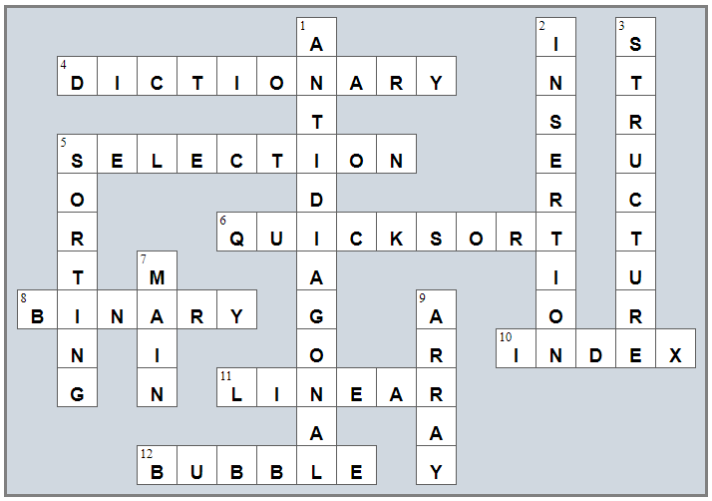
```
int countSpaces, countExistingLetters, countNonExistingLetters;
int countUserProvidedCharacters, countNonAlphabeticCharacters;
string sentence, letter;
string alphabet = "ABCDEFGHIJKLMNOPQRSTUVWXYZ";
```

```
//Create a dictionary to store the frequencies of each letter with initial
//frequencies all set to zero.
Dictionary<string, int> lettersFrequency = new();
foreach (var character in alphabet) {
  letter = "" + character;
  lettersFrequency[letter] = 0;
}

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

//Iterate through the characters of the user-provided sentence and if it is a letter,
//update (increase) the corresponding frequency count in the lettersFrequency dictionary.
//Also count number of space characters and existing letters
countSpaces = 0;
countExistingLetters = 0;
foreach (var character in sentence.ToUpper()) {
  letter = "" + character;
  if (lettersFrequency.ContainsKey(letter)) {
    lettersFrequency[letter]++;
    countExistingLetters++;
  }
  else if (letter == " ") {
    countSpaces++;
  }
}

//Display the frequency of each existing letter
foreach (var element in lettersFrequency) {
  if (element.Value > 0) {
    Console.WriteLine(element.Key + ": " + element.Value);
  }
}

//Count and display all non existing letters
countNonExistingLetters = 0;
foreach (var element in lettersFrequency) {
  if (element.Value == 0) {
    countNonExistingLetters += 1;
    Console.WriteLine(element.Key);
  }
}

//Display percentage of letters that do not exist in relation to the letters of the English alphabet
Console.WriteLine(countNonExistingLetters * 100.0 / 26 + "%");

//Display percentage of non-alphabetic characters in relation to the characters of
//the user-provided sentence (excluding space characters)
countUserProvidedCharacters = sentence.Length - countSpaces;
countNonAlphabeticCharacters = countUserProvidedCharacters - countExistingLetters;
Console.WriteLine(countNonAlphabeticCharacters * 100.0 / countUserProvidedCharacters + "%");
```

Review in “Data Structures in C#”

Review Crossword Puzzle

1.



Chapter 35

35.4 Review Questions: True/False

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

Chapter 36

36.8 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. false
18. true
19. false
20. true
21. true
22. true
23. true
24. true
25. false
26. true
27. false
28. true
29. false
30. true
31. true
32. true
33. true
34. false

36.9 Review Exercises

1. Solution

```
int findMax(int a, int b) {
  int maximum;
  if (a > b) {
    maximum = a;
  }
  else {
    maximum = b;
  }
  return maximum;
}
```

2. Solution

Step	Statement	Main Code		Method sumDigits()		
		s	i	a	d1	d2
1	s = 0	**0**	?			
2	i = 25	0	**25**			
3	i <= 27	true				
4	s += sumDigits(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 += sumDigits(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 += sumDigits(i)`			**27**	?	?
17	`d1 = a % 10`			27	**7**	?
18	`d2 = (int)(a / 10)`			27	7	**2**
19	`return d1 + d2`	**24**	27			
20	`i++`	24	**28**			
21	`i <= 27`	false				
22	`Console.WriteLine(s)`	It displays: 24				

3. Solution

Step	Statement	Main Code		Method sss()		
		s	**i**	**a**	**total**	**k**
1	`i = 1`	?	**1**			
2	`s = 0`	**0**	1			
3	`while(i < 6)`	true				
4	`if (i % 2 == 1)`	true				
5	`s += 1`	**1**	1			
6	`i++`	1	**2**			
7	`while(i < 6)`	true				
8	`if (i % 2 == 1)`	false				
9	`s += sss(i)`			**2**	?	?
10	`total = 0`			2	**0**	?
11	`k = 1`			2	0	**1**
12	`k <= a`			true		
13	`total += k`			2	**1**	1
14	`k++`			2	1	**2**
15	`k <= a`			true		
16	`total += k`			2	**3**	2

17	`k++`			2	3	**3**
18	`k <= a`				false	
19	`return total`	**4**	2			
20	`i++`	4	**3**			
21	`while(i < 6)`	true				
22	`if (i % 2 == 1)`	true				
23	`s += 1`	**5**	3			
24	`i++`	5	**4**			
25	`while(i < 6)`	true				
26	`if (i % 2 == 1)`	false				
27	`s += sss(i)`			**4**	?	?
28	`total = 0`			4	**0**	?
29	`k = 1`			4	0	**1**
30	`k <= a`				true	
31	`total += k`			4	**1**	1
32	`k++`			4	1	**2**
33	`k <= a`				true	
34	`total += k`			4	**3**	2
35	`k++`			4	3	**3**
36	`k <= a`				true	
37	`total += k`			4	**6**	4
38	`k++`			4	6	**4**
39	`k <= a`				true	
40	`total += k`			4	**10**	4
41	`k++`			4	10	**5**
42	`k <= a`				false	
43	`return total`	**15**	4			
44	`i++`	15	**5**			
45	`while(i < 6)`	true				
46	`if (i % 2 == 1)`	true				
47	`s += 1`	**16**	5			
48	`i++`	16	**6**			
49	`while(i < 6)`	false				
50	`Console.WriteLine(s)`	It displays: 16				

4. Solution

Step	Statement	Main Code				Method customDiv()	
		k	m	a	x	b	d
1	`k = Convert.ToInt32(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 + customDiv(m, a)`					2	1
7	`return (int)((b + d) / 2)`	12	2	1	**4**		
8	`Console.WriteLine(m + " " + a + " " + x)`	It displays: 2 1 4					
9	`a += 2`	12	2	**3**	4		
10	`m++`	12	**3**	3	4		
11	`while (a < 6)`	true					
12	`if (k % m != 0)`	false					
13	`x = a + m + customDiv(m, a)`					3	3
14	`return (int)((b + d) / 2)`	12	3	3	**9**		
15	`Console.WriteLine(m + " " + a + " " + x)`	It displays: 3 3 9					
16	`a += 2`	12	3	**5**	9		
17	`m++`	12	**4**	5	9		
18	`while (a < 6)`	true					
19	`if (k % m != 0)`	false					
20	`x = a + m + customDiv(m, a)`					4	5
21	`return (int)((b + d) / 2)`	12	4	5	**13**		
22	`Console.WriteLine(m + " " + a + " " + x)`	It displays: 4 5 13					
23	`a += 2`	12	4	**7**	13		
24	`m++`	12	**5**	7	13		
25	`while (a < 6)`	false					

5. Solution

Step	Statement	Main Code		void Method display()
		i	x	a
1	`i = 1`	**1**	?	
2	`i <= 5`	true		
3	`x = Convert.ToInt32(Console.ReadLine())`	1	**3**	

4	`display(x)`			**3**
5	`if (a % 2 == 0)`			false
6	`Console.WriteLine(a + " is odd")`	It displays: 3 is odd		
7	`i++`	**2**	3	
8	`i <= 5`	true		
9	`x = Convert.ToInt32(Console.ReadLine())`	2	**7**	
10	`display(x)`			**7**
11	`if (a % 2 == 0)`			false
12	`Console.WriteLine(a + " is odd")`	It displays: 7 is odd		
13	`i++`	**3**	7	
14	`i <= 5`	true		
15	`x = Convert.ToInt32(Console.ReadLine())`	3	**9**	
16	`display(x)`			**9**
17	`if (a % 2 == 0)`			false
18	`Console.WriteLine(a + " is odd")`	It displays: 9 is odd		
19	`i++`	**4**	9	
20	`i <= 5`	true		
21	`x = Convert.ToInt32(Console.ReadLine())`	4	**2**	
22	`display(x)`			**2**
23	`if (a % 2 == 0)`			true
24	`Console.WriteLine(a + " is even")`	It displays: 2 is even		
25	`i++`	**5**	2	
26	`i <= 5`	true		
27	`x = Convert.ToInt32(Console.ReadLine())`	5	**4**	
28	`display(x)`			**4**
29	`if (a % 2 == 0)`			true
30	`Console.WriteLine(a + " is even")`	It displays: 4 is even		
31	`i++`	**6**	4	
32	`i <= 5`	false		

6. Solution

Step	Statement	Main Code		void Method `division()`	
		x	y	a	b
1	`x = 20`	**20**	?		
2	`y = 30`	20	**30**		
3	`while (x % y < 30)`	true			

4	`division(y, x)`			**30**	**20**
5	`b = (int)(b / a)`			30	**0**
6	`Console.WriteLine(a * b)`	It displays: 0			
7	`x = 4 * y`	**120**	30		
8	`y++`	120	**31**		
9	`while (x % y < 30)`	true			
10	`division(y, x)`			**31**	**120**
11	`b = (int)(b / a)`			31	**3**
12	`Console.WriteLine(a * b)`	It displays: 93			
13	`x = 4 * y`	**124**	31		
14	`y++`	124	**32**		
15	`while (x % y < 30)`	true			
16	`division(y, x)`			**32**	**124**
17	`b = (int)(b / a)`			32	**3**
18	`Console.WriteLine(a * b)`	It displays: 96			
19	`x = 4 * y`	**128**	32		
20	`y++`	128	**33**		
21	`while (x % y < 30)`	true			
22	`division(y, x)`			**33**	**128**
23	`b = (int)(b / a)`			33	**3**
24	`Console.WriteLine(a * b)`	It displays: 99			
25	`x = 4 * y`	**132**	33		
26	`y++`	132	**34**		
27	`while (x % y < 30)`	false			

7. Solution

Step	Statement	Main Code		void Method `calculate()`		
		i	m	n	s	j
1	`i = 1`	**1**	?			
2	`i <= 3`	true				
3	`m = Convert.ToInt32(Console.ReadLine())`	1	**2**			
4	`calculate(m)`			**2**	?	?
5	`s = 0`			2	**0**	?
6	`j = 2`			2	0	**2**
7	`j <= 2 * n`			true		
8	`s = s + Math.Pow(j, 2)`			2	**4**	2

9	`j += 2`			2	4	**4**
10	`j <= 2 * n`				true	
11	`s = s + Math.Pow(j, 2)`			2	**20**	4
12	`j += 2`			2	20	**6**
13	`j <= 2 * n`				false	
14	`Console.WriteLine(s)`	It displays: 20				
15	`i++`	**2**	2			
16	`i <= 3`	true				
17	`m = Convert.ToInt32(Console.ReadLine())`	2	**3**			
18	`calculate(m)`			**3**	?	?
19	`s = 0`			3	**0**	?
20	`j = 2`			3	0	**2**
21	`j <= 2 * n`				true	
22	`s = s + Math.Pow(j, 2)`			3	**4**	2
23	`j += 2`			3	4	**4**
24	`j <= 2 * n`				true	
25	`s = s + Math.Pow(j, 2)`			3	**20**	4
26	`j += 2`			3	20	**6**
27	`j <= 2 * n`				true	
28	`s = s + Math.Pow(j, 2)`			3	**56**	6
29	`j += 2`			3	56	**8**
30	`j <= 2 * n`				false	
31	`Console.WriteLine(s)`	It displays: 56				
32	`i++`	**3**	3			
33	`i <= 3`	true				
34	`m = Convert.ToInt32(Console.ReadLine())`	3	**4**			
35	`calculate(m)`			**4**	?	?
36	`s = 0`			4	**0**	?
37	`j = 2`			4	0	**2**
38	`j <= 2 * n`				true	
39	`s = s + Math.Pow(j, 2)`			4	**4**	2
40	`j += 2`			4	4	**4**
41	`j <= 2 * n`				true	
42	`s = s + Math.Pow(j, 2)`			4	**20**	4
43	`j += 2`			4	20	**6**
44	`j <= 2 * n`				true	
45	`s = s + Math.Pow(j, 2)`			4	**56**	6

46	`j += 2`			4	56	**8**
47	`j <= 2 * n`				true	
48	`s = s + Math.Pow(j, 2)`			4	**120**	8
49	`j += 2`			4	120	**10**
50	`j <= 2 * n`				false	
51	`Console.WriteLine(s)`	It displays: 120				
52	`i++`	**4**	4			
53	`i <= 3`	false				

8. Solution

```
int findSum(int a, int b, int c) {
  return a + b + c;
}
```

9. Solution

```
double findAvg(double a, double b, double c, double d) {
  return (a + b + c + d) / 4;
}
```

10. Solution

```
double maximum(double a, double b, double c) {
  double m;

  m = a;
  if (b > m) {
    m = b;
  }
  if (c > m) {
    m = c;
  }
  return m;
}
```

11. Solution

```
void displayMax(double a, double b, double c, double d, double e) {
  double m;

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

```
    m = d;
  }
  if (e > m) {
    m = e;
  }
  Console.WriteLine(m);
}
```

12. Solution

```
double myRound(double x) {
  int digitToCheck;
  double returnValue;

  digitToCheck = (int)(x * 1000) % 10;
  if (digitToCheck >= 5) {
    returnValue = ((int)(x * 100) + 1) / 100.0;
  }
  else {
    returnValue = ((int)(x * 100)) / 100.0;
  }

  return returnValue;
}
```

13. Solution

```
double findMin(double a, double b) {
  double minimum;

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

double temp1, temp2, x1, x2, x3, x4;

Console.WriteLine("Enter four numbers: ");
x1 = Convert.ToDouble(Console.ReadLine());
x2 = Convert.ToDouble(Console.ReadLine());
x3 = Convert.ToDouble(Console.ReadLine());
x4 = Convert.ToDouble(Console.ReadLine());

//First approach
temp1 = findMin(x1, x2);
temp2 = findMin(x3, x4);
Console.WriteLine(findMin(temp1, temp2));

//Second approach
Console.WriteLine(findMin(findMin(x1, x2), findMin(x3, x4)));
```

14. Solution

```
double KelvinToFahrenheit(double kelvin) {
  return 1.8 * kelvin - 459.67;
}

double KelvinToCelsius(double kelvin) {
  return kelvin - 273.15;
}

double k;

Console.Write("Enter a temperature in degrees Kelvin: ");
k = Convert.ToDouble(Console.ReadLine());
Console.WriteLine("Fahrenheit: " + KelvinToFahrenheit(k));
Console.WriteLine("Celsius: " + KelvinToCelsius(k));
```

15. Solution

```
string bmi(double w, double h) {
  double b;
  string returnValue;

  b = w * 703 / Math.Pow(h, 2);
  if (b < 16) {
    returnValue = "You must add weight.";
  }
  else if (b < 18.5) {
    returnValue = "You should add some weight.";
  }
  else if (b < 25) {
    returnValue = "Maintain your weight.";
  }
  else if (b < 30) {
    returnValue = "You should lose some weight.";
  }
  else {
    returnValue = "You must lose weight.";
  }

  return returnValue;
}

double height, weight;
int age;

Console.Write("Enter your weight (in pounds): ");
weight = Convert.ToDouble(Console.ReadLine());
while (weight < 0) {
  Console.Write("Error! Enter your weight (in pounds): ");
  weight = Convert.ToDouble(Console.ReadLine());
}
```

```
Console.Write("Enter your age: ");
age = Convert.ToInt32(Console.ReadLine());
while (age < 18) {
  Console.Write("Error! Enter your age: ");
  age = Convert.ToInt32(Console.ReadLine());
}

Console.Write("Enter your height (in inches): ");
height = Convert.ToDouble(Console.ReadLine());
while (height < 0) {
  Console.Write("Error! Enter your height (in inches): ");
  height = Convert.ToDouble(Console.ReadLine());
}

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

16. Solution

```
void numOfDays(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);
}

int m, y;

Console.Write("Enter a year: ");
y = Convert.ToInt32(Console.ReadLine());
for (m = 1; m <= 12; m++) {
  numOfDays(y, m);
}
```

17. Solution

```
int numOfDays(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;
  }

  return days;
}

int y, m1, m2, m, total;

Console.Write("Enter a year: ");
y = Convert.ToInt32(Console.ReadLine());
Console.Write("Enter a month: ");
m1 = Convert.ToInt32(Console.ReadLine());
Console.Write("Enter a second month: ");
m2 = Convert.ToInt32(Console.ReadLine());

total = 0;
for (m = m1; m <= m2; m++) {
  total += numOfDays(y, m);
}
Console.WriteLine(total);
```

18. Solution

```
void displayMenu() {
  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: ");
}
```

```
void metersToMiles(double meters) {
  Console.WriteLine(meters + " meters equals " + (meters / 1609.344) + " miles");
}

void milesToMeters(double miles) {
  Console.WriteLine(miles + " miles equals " + (miles * 1609.344) + " meters");
}

int choice;
double distance;

displayMenu();
choice = Convert.ToInt32(Console.ReadLine());
while (choice != 3) {
  Console.Write("Enter distance: ");
  distance = Convert.ToInt32(Console.ReadLine());
  if (choice == 1) {
    metersToMiles(distance);
  }
  else {
    milesToMeters(distance);
  }

  displayMenu();
  choice = Convert.ToInt32(Console.ReadLine());
}
```

19. Solution

```
void amountToPay(int seconds) {
  double extra, tax, total, totalWithoutTax;

  if (seconds <= 600) {
    extra = 0;
  }
  else if (seconds <= 1200) {
    extra = (seconds - 600) * 0.01;
  }
  else {
    extra = 600 * 0.01 + (seconds - 1200) * 0.02;
  }

  totalWithoutTax = 10 + extra;
  tax = totalWithoutTax * 11 / 100;
  total = totalWithoutTax + tax;

  Console.WriteLine("Total amount to pay: " + total);
}

int seconds;

Console.Write("Enter number of seconds: ");
seconds = Convert.ToInt32(Console.ReadLine());
```

```
amountToPay(seconds);
```

Chapter 37

37.9 Review Questions: True/False

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

37.10 Review Exercises

1. Solution

It displays: 5

2. Solution

It displays: 14

3. Solution

It displays: 14

4. Solution

It displays: hellohellohello

5. Solution

It displays: 15

6. Solution

It displays: 11 4

7. Solution

It displays: 3

8. Solution

Within the method `getNumOfDigits()`, variable `x` eventually becomes 0, and since the variable `val` is passed to the method by reference, that 0 also reflects back to the main code. So, when the flow of execution returns to the main code, the value of variable `val` is zeroed!

To resolve this issue, all you have to do is remove the `ref` keyword at the beginning of the formal arguments `x` and `val`. If you do so, the variable `val` is passed to the method by value, so that no matter what happens to variable `x` within the method, nothing can affect the value of the variable `val` of the main code.

9. Solution

```
const int STUDENTS = 10;
const int LESSONS = 5;

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] = Convert.ToInt32(Console.ReadLine());
    }
  }
}

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;
}

void part3(double[] average, string[] names) {
  int m, n;
  double temp;
  string tempStr;

  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;

        tempStr = names[n];
        names[n] = names[n - 1];
        names[n - 1] = tempStr;
      }
      else if (average[n] == average[n - 1]) {
```

```
        if (names[n].CompareTo(names[n - 1]) < 0) {
          tempStr = names[n];
          names[n] = names[n - 1];
          names[n - 1] = tempStr;
        }
      }
    }
  }
}

int i;

string[] names = new string[STUDENTS];
int[,] grades = new int[STUDENTS, LESSONS];
double[] average = new double[STUDENTS];

part1(names, grades);

average = part2(grades);

part3(average, names);

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

10. Solution

```
string part1() {
  string message;

  Console.Write("Enter a message: ");
  message = Console.ReadLine().ToLower();
  return message;
}

string part2(string message) {
  string letter, messageClean;
  int i;

  messageClean = "";
  for (i = 0; i <= message.Length - 1; i++) {
    letter = "" + message[i];
    if (letter != " " && letter != "," && letter != "." && letter != "?") {
      messageClean += letter;
    }
  }
  return messageClean;
}

bool part3(string messageClean) {
  int middlePos, i, j;
  bool palindrome;
  string leftLetter, rightLetter;
```

```
  middlePos = (int)((messageClean.Length - 1) / 2);
  j = messageClean.Length - 1;
  palindrome = true;
  for (i = 0; i <= middlePos; i++) {
    leftLetter = "" + messageClean[i];
    rightLetter = "" + messageClean[j];
    if (leftLetter != rightLetter) {
      palindrome = false;
      break;
    }
    j--;
  }
  return palindrome;
}

bool part4(string message) {
  string messageClean;
  bool palindrome;

  messageClean = part2(message);
  palindrome = part3(messageClean);
  return palindrome;
}

string message;
bool palindrome;

message = part1();
palindrome = part4(message);
if (palindrome) {
  Console.WriteLine("The message is palindrome");
}
```

11. Solution

```
int a, b, c, d, maximum;

a = Convert.ToInt32(Console.ReadLine());
b = Convert.ToInt32(Console.ReadLine());
c = Convert.ToInt32(Console.ReadLine());
d = Convert.ToInt32(Console.ReadLine());

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

```
Console.WriteLine(maximum);
```

12. Solution

First approach

```
void f1(double a, double b, double c, ref double total, ref double average) {
  total = a + b + c;
  average =  total / 3;
}
```

Second approach

```
void f1(double a, double b, double c, double[] returningArray) {
  returningArray[0] = a + b + c;
  returningArray[1] =  returningArray[0] / 3;
}
```

13. Solution

```
double myRound(double x, int decimalPlaces = 2) {
  double returnValue;

  int digitToCheck = (int)((x * Math.Pow(10, decimalPlaces + 1))) % 10;
  if (digitToCheck >= 5) {
    returnValue = ((int)((x * Math.Pow(10, decimalPlaces))) + 1) / Math.Pow(10, decimalPlaces);
  }
  else {
    returnValue = ((int)(x * Math.Pow(10, decimalPlaces))) / Math.Pow(10, decimalPlaces);
  }
  return returnValue;
}
```

14. Solution

```
bool getInput() {
  string answer;

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

  return answer == "YES";  //This returns true or false
}

double findArea(double b, double h) {
  return b * h;
}

double b, h;

do {
  Console.Write("Enter the base of the parallelogram: ");
  b = Convert.ToDouble(Console.ReadLine());
```

```
  Console.Write("Enter the height of the parallelogram: ");
  h = Convert.ToDouble(Console.ReadLine());

  Console.WriteLine("Area = " + findArea(b, h));

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

15. Solution

```
const int STUDENTS = 100;

void getArrays(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] = Convert.ToInt32(Console.ReadLine());
  }
}

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

void sortArrays(int[] grades, string[] names) {
  int m, n, elementGrds;
  string elementNms;

  for (m = 1; m <= STUDENTS - 1; m++) {
    elementGrds = grades[m];
    elementNms = names[m];

    n = m;
    while (n > 0 && grades[n - 1] > elementGrds) {
      grades[n] = grades[n - 1];
      names[n] = names[n - 1];
      n--;
    }

    grades[n] = elementGrds;
    names[n] = elementNms;
  }
}

int i;
double average;
```

```
string[] names = new string[STUDENTS];
int[] grades = new int[STUDENTS];

getArrays(names, grades);
average = getAverage(grades);
sortArrays(grades, names);
for (i = 0; i <= STUDENTS - 1; i++) {
  if (grades[i] < average) {
    Console.WriteLine(names[i]);
  }
}
```

16. Solution

```
const int JUDGES = 10;

int[] getArray() {
  int[] score = new int[JUDGES];
  int i;

  for (i = 0; i <= JUDGES - 1; i++) {
    Console.Write("Judge No " + (i + 1) + ". Enter score: ");
    score[i] = Convert.ToInt32(Console.ReadLine());
  }
  return score;
}

void findMinMax(int[] score, ref int minimum, ref int maximum) {
  int i;
  minimum = score[0];
  maximum = score[0];
  for (i = 1; i <= JUDGES - 1; i++) {
    if (score[i] > maximum) {
      maximum = score[i];
    }
    if (score[i] < minimum) {
      minimum = score[i];
    }
  }
}

string name;
int total, i, points, minimum = 0, maximum = 0;

Console.Write("Enter artist's name: ");
name = Console.ReadLine();
int[] score = getArray();
findMinMax(score, ref minimum, ref maximum);

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

```
points = total - minimum - maximum;
Console.WriteLine("Artist " + name + " got " + points + " points");
```

17. Solution

```
int sumRecursive(int n) {
  if (n == 1) {
    return 1;
  }
  else {
    return sumRecursive(n - 1) + n;
  }
}

//Main code starts here
int num = Convert.ToInt32(Console.ReadLine());
Console.WriteLine(sumRecursive(num));
```

18. Solution

```
double woc(int index) {
  if (index == 1) {
    return 1;
  }
  else {
    return 2 * woc(index - 1);
  }
}

double total;
int i;

total = 0;
for (i = 1; i <= 64; i++) {
  total += woc(i);
}
Console.WriteLine(total);
```

19. Solution

```
int fib(int n) {
  if (n <= 1) {
    return n;
  }
  else {
    return fib(n - 1) + fib(n - 2);
  }
}

//Main code starts here
int num = Convert.ToInt32(Console.ReadLine());
Console.WriteLine(fib(num - 1));
```

20. Solution

```
int tribonacci(int n) {
  if (n == 0) {
    return 0;
  }
  else if (n == 1 || n == 2) {
    return 1;
  }
  else {
    return tribonacci(n - 1) + tribonacci(n - 2) + tribonacci(n - 3);
  }
}
```

21. Solution

```
double myPow(double n, int p) {
  if (p == 0)
    return 1;
  else if (p < 0)
    return 1 / (n * myPow(n, -p - 1));
  else
    return n * myPow(n, p - 1);
}
```

22. Solution

```
double factorial(int value) {
  if (value == 1) {
    return 1;
  }
  else {
    return value * factorial(value - 1);
  }
}

double myCos(double x, int i = 40) {
  if (i == 0) {
    return 1;
  }
  else {
    return myCos(x, i - 4) + Math.Pow(x, i) / factorial(i) - Math.Pow(x, i - 2) / factorial(i - 2);
  }
}

Console.WriteLine(myCos(Math.PI / 4));
```

Chapter 38

38.3 Review Exercises

1. Solution

```
void displayMenu() {
  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.WriteLine("5. Exit");
  Console.WriteLine("-------------------------");
  Console.Write("Enter a choice: ");
}

double USD_to_EU(double value) {
  return value * 0.94;
}

double USD_to_GBP(double value) {
  return value * 0.81;
}

double USD_to_JPY(double value) {
  return value * 149.11;
}

double USD_to_CAD(double value) {
  return value * 1.36;
}

int choice;
double amount;

displayMenu();
choice = Convert.ToInt32(Console.ReadLine());
while (choice != 5) {
  Console.Write("Enter an amount in US dollars: ");
  amount = Convert.ToDouble(Console.ReadLine());
  switch (choice) {
    case 1:
      Console.WriteLine(amount + " USD = " + USD_to_EU(amount) + " Euro");
      break;
    case 2:
      Console.WriteLine(amount + " USD = " + USD_to_GBP(amount) + " GBP");
      break;
    case 3:
      Console.WriteLine(amount + " USD = " + USD_to_JPY(amount) + " JPY");
      break;
    case 4:
      Console.WriteLine(amount + " USD = " + USD_to_CAD(amount) + " CAD");
```

```
      break;
  }

  displayMenu();
  choice = Convert.ToInt32(Console.ReadLine());
}
```

2. Solution

```
void displayMenu() {
  Console.WriteLine("---------------------------------------------");
  Console.WriteLine("1. Convert USD to Euro (EUR)");
  Console.WriteLine("2. Convert USD to British Pound Sterling (GBP)");
  Console.WriteLine("3. Convert EUR to USD");
  Console.WriteLine("4. Convert EUR to GBP");
  Console.WriteLine("5. Convert GBP to USD");
  Console.WriteLine("6. Convert GBP to EUR");
  Console.WriteLine("7. Exit");
  Console.WriteLine("-------------------------");
  Console.Write("Enter a choice: ");
}

double USD_to_EUR(double value) {
  return value * 0.94;
}

double USD_to_GBP(double value) {
  return value * 0.81;
}

int choice;
double amount;

displayMenu();
choice = Convert.ToInt32(Console.ReadLine());
while (choice != 7) {
  Console.Write("Enter an amount: ");
  amount = Convert.ToDouble(Console.ReadLine());
  switch (choice) {
    case 1:
      Console.WriteLine(amount + " USD = " + USD_to_EUR(amount) + " Euro");
      break;
    case 2:
      Console.WriteLine(amount + " USD = " + USD_to_GBP(amount) + " GBP");
      break;
    case 3:
      Console.WriteLine(amount + " EUR = " + 1 / USD_to_EUR(1 / amount) + " USD");
      break;
    case 4:
      Console.WriteLine(amount + " EUR = " + USD_to_GBP(1 / USD_to_EUR(1 / amount)) + " GBP");
      break;
    case 5:
```

```
      Console.WriteLine(amount + " GBP = " + 1 / USD_to_GBP(1 / amount) + " USD");
      break;
    case 6:
      Console.WriteLine(amount + " GBP = " + USD_to_EUR(1 / USD_to_GBP(1 / amount)) + " EUR");
      break;
  }

  displayMenu();
  choice = Convert.ToInt32(Console.ReadLine());
}
```

3. Solution

```
const double ACCURACY = 0.000000001;

double factorial(int n) {
  int i;

  double returnValue = 1;
  for (i = 1; i <= n; i++) {
    returnValue *= i;
  }
  return returnValue;
}

double mySin(double x) {
  int i, sign;
  double sinus, sinusPrevious;
  sign = 1;
  sinus = 0;
  i = 1;
  do {
    sinusPrevious = sinus;
    sinus += sign * Math.Pow(x, i) / factorial(i);

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

double degreesToRad(double degrees) {
  return 2 * Math.PI * degrees / 360;
}

int i;

for (i = 0; i <= 360; i++) {
  Console.WriteLine("sin(" + i + ") ~= " + mySin(degreesToRad(i)));
}
```

4. Solution

```
bool isLeap(int year) {
  bool returnValue = false;
  if (year % 4 == 0 && year % 100 != 0 || year % 400 == 0) {
    returnValue = true;
  }
  return returnValue;
}

int numOfDays(int year, int month) {
  int days;

  switch (month) {
    case 4:
    case 6:
    case 9:
    case 11:
      days = 30;
      break;
    case 2:
      if (isLeap(year)) {
        days = 29;
      }
      else {
        days = 28;
      }
      break;
    default:
      days = 31;
      break;
  }

  return days;
}

bool checkDate(int day, int month, int year) {
  bool returnValue = true;
  if (month < 1 || month > 12) {
    returnValue = false;
  }
  else if (day < 1 || day > numOfDays(year, month)) {
    returnValue = false;
  }
  return returnValue;
}

int day, month, year, total, i;

Console.Write("Enter day: ");
day = Convert.ToInt32(Console.ReadLine());
Console.Write("Enter month: ");
```

```
month = Convert.ToInt32(Console.ReadLine());
Console.Write("Enter year: ");
year = Convert.ToInt32(Console.ReadLine());
while (!checkDate(day, month, year)) {
  Console.WriteLine("Error!");
  Console.Write("Enter day: ");
  day = Convert.ToInt32(Console.ReadLine());
  Console.Write("Enter month: ");
  month = Convert.ToInt32(Console.ReadLine());
  Console.Write("Enter year: ");
  year = Convert.ToInt32(Console.ReadLine());
}

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

Console.WriteLine(total);
```

5. Solution

```
int dice() {
  Random rnd = new();
  return rnd.Next(1, 7);
}

int dice1, dice2, i, player, total, totalPlayer1 = 0, totalPlayer2 = 0;
string[] names = new string[2];

Console.Write("Player1 - Enter name: ");
names[0] = Console.ReadLine();
Console.Write("Player2 - Enter name: ");
names[1] = Console.ReadLine();

for (player = 0; player <= 1; player++) {
  total = 0;
  for (i = 1; i <= 10; i++) {
    Console.WriteLine(names[player] + ", hit enter to roll the dice!");
    Console.ReadLine();  //This statement just waits the user to hit the enter key

    dice1 =  dice();
    dice2 =  dice();
    Console.WriteLine(dice1 + "   " + dice2);
    total += dice1 + dice2;
  }
  if (player == 1) {
    totalPlayer1 = total;
  }
  else {
    totalPlayer2 = total;
  }
```

```
}

if (totalPlayer1 == totalPlayer2) {
  Console.WriteLine("Tie!");
}
else if (totalPlayer1 > totalPlayer2) {
  Console.WriteLine(names[0] + " wins");
}
else {
  Console.WriteLine(names[1] + " wins");
}
```

6. Solution

```
const int GAS = 1;
const int DIESEL = 2;
const int HYBRID = 3;
const double TAX_RATE = 0.10;
const int CARS = 40;

int getChoice() {
  Console.WriteLine("1. Gas");
  Console.WriteLine("2. Diesel");
  Console.WriteLine("3. Hybrid");
  Console.Write("Enter type of the car: ");
  return Convert.ToInt32(Console.ReadLine());
}

int getDays() {
  Console.Write("Enter total number of rental days: ");
  return Convert.ToInt32(Console.ReadLine());
}

double getCharge(int carType, int rentalDays) {
  double charge;

  if (carType == GAS) {
    if (rentalDays <= 5) {
      charge = rentalDays * 24;
    }
    else if (rentalDays <= 8) {
      charge = 5 * 24 + (rentalDays - 5) * 22;
    }
    else {
      charge = 5 * 24 + 3 * 22 + (rentalDays - 8) * 18;
    }
  }
  else if (carType == DIESEL) {
    if (rentalDays <= 5) {
      charge = rentalDays * 28;
    }
    else if (rentalDays <= 8) {
```

```
      charge = 5 * 28 + (rentalDays - 5) * 25;
    }
    else {
      charge = 5 * 28 + 3 * 25 + (rentalDays - 8) * 21;
    }
  }
  else {
    if (rentalDays <= 5) {
      charge = rentalDays * 30;
    }
    else if (rentalDays <= 8) {
      charge = 5 * 30 + (rentalDays - 5) * 28;
    }
    else {
      charge = 5 * 30 + 3 * 28 + (rentalDays - 8) * 23;
    }
  }
  charge = charge * (1 + TAX_RATE); //This is equivalent to charge += charge * TAX_RATE;
  return charge;
}

int count, i;
double charge, total;

int[] rentedCarTypes = new int[CARS];
int[] rentedDays = new int[CARS];

for (i = 0; i <= CARS - 1; i++) {
  rentedCarTypes[i] = getChoice();
  rentedDays[i] = getDays();
}

total = 0;
for (i = 0; i <= CARS - 1; i++) {
  charge = getCharge(rentedCarTypes[i], rentedDays[i]);
  Console.WriteLine("Car No " + (i + 1) + ": " + charge);
  total += charge;
}

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

Console.WriteLine("Hybrids rented: " + count);
Console.WriteLine("Net profit: " + total / (1 + TAX_RATE));
```

7. Solution

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

```
void getData(string[] names, int[,] viewers) {
  int i, j;
  string[] dayNames = {"Monday", "Tuesday", "Wednesday", "Thursday", "Friday", "Saturday", "Sunday"};

  for (i = 0; i <= CHANNELS - 1; i++) {
    Console.Write("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 " + dayNames[j] +
                    " for channel " + names[i] + ": ");
      viewers[i, j] = Convert.ToInt32(Console.ReadLine());
    }
  }
}

double getAverage(int[] a) {
  int total ,i;

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

int i, j;
double weekend;
bool increasing;

string[] names = new string[CHANNELS];
int[,] viewers = new int[CHANNELS, DAYS];
getData(names, viewers);

int[] temporaryArray = new int[5];
for (i = 0; i <= CHANNELS - 1; i++) {
  for (j = 0; j <= 4; j++) {
    temporaryArray[j] = viewers[i, j];
  }
  weekend = (viewers[i, DAYS - 2] + viewers[i, DAYS - 1]) / 2;
  if (weekend >= 1.2 * getAverage(temporaryArray)) {
    Console.WriteLine(names[i]);
  }
}

for (i = 0; i <= CHANNELS - 1; i++) {
  increasing = true;
  for (j = 1; j <= DAYS - 1; j++) {
    if (viewers[i, j] <= viewers[i, j - 1]) {
      increasing = false;
    }
  }
  if (increasing) {
```

```
    Console.WriteLine(names[i]);
  }
}
```

8. Solution

```
const int CITIZENS = 300;

void inputData(long[] SSNs, string[] answers) {
  int i;

  for (i = 0; i <= CITIZENS - 1; i++) {
    Console.Write("Enter SSN: ");
    SSNs[i] = Int64.Parse(Console.ReadLine());
    Console.Write("Enter answer: ");
    answers[i] = Console.ReadLine();
  }
}

void sortArrays(long[] SSNs, string[] answers) {
  int m, n, indexOfMin;
  long minimum, temp;
  string tempStr;

  for (m = 0; m <= CITIZENS - 1; m++) {
    minimum = SSNs[m];
    indexOfMin = m;
    for (n = m; n <= CITIZENS - 1; n++) {
      if (SSNs[n] < minimum) {
        minimum = SSNs[n];
        indexOfMin = n;
      }
    }
    temp = SSNs[m];
    SSNs[m] = SSNs[indexOfMin];
    SSNs[indexOfMin] = temp;
    tempStr = answers[m];
    answers[m] = answers[indexOfMin];
    answers[indexOfMin] = tempStr;
  }
}

int searchArray(long[] SSNs, long SSN) {
  int left, right, middle, returnValue;
  bool found;

  middle = -1
  left = 0;
  right = CITIZENS - 1;
  found = false;
  while (left <= right && !found) {
    middle = (int)((left + right) / 2);
```

```
    if (SSN < SSNs[middle]) {
      right = middle - 1;
    }
    else if (SSN > SSNs[middle]) {
      left = middle + 1;
    }
    else {
      found = true;
    }
  }

  if (!found) {
    Console.WriteLine("SSN not found!");
    return -1;
  }
  else {
    return middle;
  }
}

int countAnswers(string[] answers, string answer) {
  int count, i;

  count = 0;
  for (i = 0; i <= CITIZENS - 1; i++) {
    if (answers[i] == answer) {
      count++;
    }
  }
  return count;
}

long[] SSNs = new long[CITIZENS];
long SSN;
string[] answers = new string[CITIZENS];
int index, count;
string answer;

do {
  inputData(SSNs, answers);
  sortArrays(SSNs, answers);

  Console.Write("Enter an SSN to search: ");
  SSN = Int64.Parse(Console.ReadLine());

  index = searchArray(SSNs, SSN);
  if (index != -1) {
    answer = answers[index];
    Console.WriteLine(answer);

    count = countAnswers(answers, answer);
    Console.WriteLine(count * 100 / (double)CITIZENS);
  }
```

```
  Console.Write("Repeat? ");
  answer = Console.ReadLine();
} while (answer == "yes");
```

9. Solution

```
const int TEAMS = 8;
const int GAMES = 12;

void inputData(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();
    }
  }
}

void displayResult(string[] names, string[,] results) {
  string result;
  int i, j;
  bool found;

  Console.Write("Enter a result to search (W, L, T): ");
  result = Console.ReadLine();
  for (i = 0; i <= TEAMS - 1; i++) {
    Console.WriteLine("Team: " + names[i]);
    found = false;
    for (j = 0; j <= GAMES - 1; j++) {
      if (results[i, j] == result) {
        Console.WriteLine("Week: " + (j + 1));
        found = true;
      }
    }
    if (!found) {
      Console.WriteLine("Nothing found");
    }
  }
}

int findTeam(string[] names) {
  string name;
  int i, returnValue;

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

  i = 0;
  while (i < TEAMS - 1 && names[i] != name) {
```

```
    i++;
  }

  if (names[i] != name) {
    returnValue = -1;
  }
  else {
    returnValue = i;
  }
  return returnValue;
}

string[] names = new string[TEAMS];
string[,] results = new string[TEAMS, GAMES];
int j, index, total;

inputData(names, results);
displayResult(names, results);

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

if (index == -1) {
  Console.WriteLine("Team not found");
}
```

10. Solution

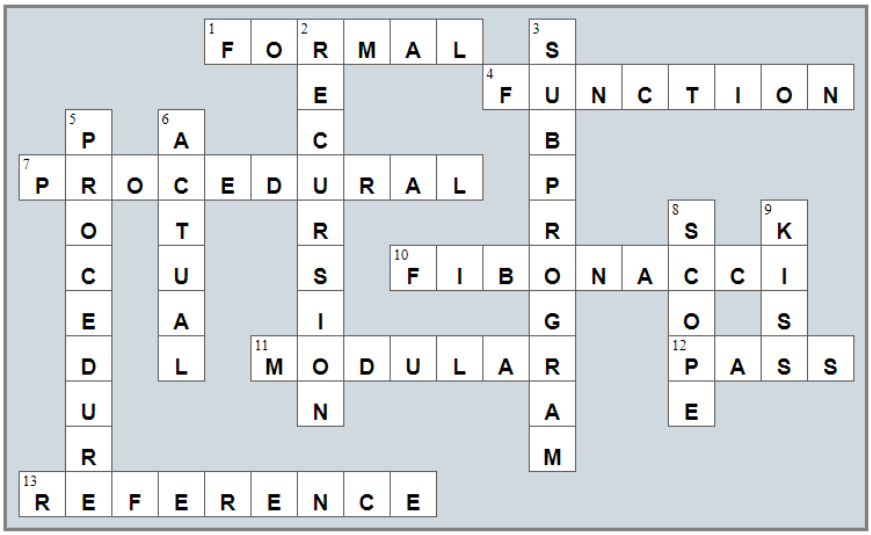
```
bool hasDuplicateDigits(int num) {
  //Initialize an array to store the count of each digit
  int[] digitCount = {0, 0, 0, 0, 0, 0, 0, 0, 0, 0};

  while (num > 0) {
    int digit = num % 10; //Extract the last digit
    if (digitCount[digit] > 0) {
      return true; //If this digit has been seen before, return true
    }
    digitCount[digit]++; //Increment the count of this digit
    x = (int)(x / 10);   //Move to the next digit
  }
```

```
  return false; //No duplicate digits found
}

int num;

Console.Write("Enter an integer: ");
num = Convert.ToInt32(Console.ReadLine());
while (num < 11) {
  Console.Write("Wrong number! Enter an integer greater than 10: ");
  num = Convert.ToInt32(Console.ReadLine());
}

if (hasDuplicateDigits(num)) {
  Console.WriteLine("The integer contains duplicate digits");
}
else {
  Console.WriteLine("The integer does not contain duplicate digits");
}
```

Review in “Subprograms”

Review Crossword Puzzle

1.



Chapter 39

39.8 Review Questions: True/False

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

39.9 Review Exercises

1. Solution

```
double sqrSide, rctnglBase, rctnglHeight, trnglBase, trnglHeight;
Geometry gmtr = new();

Console.Write("Enter square side: ");
sqrSide = Convert.ToDouble(Console.ReadLine());

Console.Write("Enter rectangle base:  ");
rctnglBase = Convert.ToDouble(Console.ReadLine());
Console.Write("Enter rectangle height: ");
rctnglHeight = Convert.ToDouble(Console.ReadLine());

Console.Write("Enter triangle base: ");
trnglBase = Convert.ToDouble(Console.ReadLine());
Console.Write("Enter triangle height: ");
trnglHeight = Convert.ToDouble(Console.ReadLine());

Console.WriteLine(gmtr.rectangleArea(sqrSide, sqrSide));
Console.WriteLine(gmtr.rectangleArea(rctnglBase, rctnglHeight));
Console.WriteLine(gmtr.triangleArea(trnglBase, trnglHeight));

class Geometry {
  public double rectangleArea(double b, double h) {
    return b * h;
  }

  public double triangleArea(double b, double h) {
    return b * h / 2;
  }
}
```

2. Solution

```
Pet pet1 = new();
pet1.kind = "dog";
pet1.legsNumber = 4;

Pet pet2 = new();
```

```
pet2.kind = "monkey";
pet2.legsNumber = 2;

pet1.startRunning();
pet2.startRunning();
pet1.stopRunning();

class Pet {
  public string kind;
  public int legsNumber;

  public void startRunning() {
    Console.WriteLine("Pet is running");
  }

  public void stopRunning() {
    Console.WriteLine("Pet stopped");
  }
}
```

3. Solution

```
Pet pet1 = new("dog", 4);

pet1.startRunning();
pet1.stopRunning();

pet1.Kind = ""; //This will throw an error
pet1.LegsNumber = -1; //This will throw an error

class Pet {
  private string _kind;
  private int _legsNumber;

  //Define the constructor
  public Pet(string kind, int legsNumber) {
    //Initialize fields (using the corresponding properties)
    this.Kind = kind;
    this.LegsNumber = legsNumber;
  }

  //Define public property Kind
  public string Kind {
    get {
      return this._kind;
    }

    set {
      if (value != "") {
        this._kind = value;
      }
      else {
        throw new Exception("Cannot be empty");
```

```
      }
    }
  }

  //Define public property LegsNumber
  public int LegsNumber {
    get {
      return this._legsNumber;
    }

    set {
      if (value >= 0) {
        this._legsNumber = value;
      }
      else {
        throw new Exception("Cannot be negative");
      }
    }
  }

  public void startRunning() {
    Console.WriteLine("Pet is running");
  }

  public void stopRunning() {
    Console.WriteLine("Pet stopped");
  }
}
```

4. Solution

```
const int BOXES = 30;

int i;
double w, l, h;

Box[] listOfObj = new Box[BOXES]; //create an array

for (i = 0; i <= BOXES - 1; i++) {
  Console.Write("Enter width: ");
    w = Convert.ToDouble(Console.ReadLine());
    Console.Write("Enter length: ");
    l = Convert.ToDouble(Console.ReadLine());
    Console.Write("Enter height: ");
    h = Convert.ToDouble(Console.ReadLine());

    //Add each new object to the array
    listOfObj[i] = new(w, l, h);
}

for (i = 0; i <= BOXES - 1; i++) {
    listOfObj[i].displayDimensions();
    listOfObj[i].displayVolume();
```

```
}

class Box {
  private double _width;
  private double _length;
  private double _height;

  //Define the constructor
  public Box(double w, double l, double h) {
    //Initialize fields
    this._width = w;
    this._length = l;
    this._height = h;
  }

  public void displayVolume() {
    Console.WriteLine("Volume: " + (this._width * this._length * this._height));
  }

  public void displayDimensions() {
    Console.WriteLine(this._width + " x " + this._length + " x " + this._height);
  }
}
```

5. Solution

```
const int BOXES = 30;

int i;
double w, l, h;

Box[] listOfObj = new Box[BOXES]; //Create an array

for (i = 0; i <= BOXES - 1; i++) {
  Console.Write("Enter width: ");
  w = Convert.ToDouble(Console.ReadLine());
  Console.Write("Enter length: ");
  l = Convert.ToDouble(Console.ReadLine());
  Console.Write("Enter height: ");
  h = Convert.ToDouble(Console.ReadLine());

  //Add each new object to the array
  listOfObj[i] = new(w, l, h);
}

for (i = 0; i <= BOXES - 1; i++) {
  listOfObj[i].displayDimensions();
  listOfObj[i].displayVolume();
}

class Box {
  private double _width;
  private double _length;
```

```
private double _height;

//Define the constructor
public Box(double w, double l, double h) {
  //Initialize fields (using the corresponding properties)
  this.Width = w;
  this.Length = l;
  this.Height = h;
}

//Define public property Width
public double Width {
  //Define the getter
  get {
    return this._width;
  }

  //Define the setter
  set {
    if (value > 0) {
      this._width = value;
    }
    else {
      throw new Exception("Cannot be negative or zero");
    }
  }
}

//Define public property Length
public double Length {
  //Define the getter
  get {
    return this._length;
  }

  //Define the setter
  set {
    if (value > 0) {
      this._length = value;
    }
    else {
      throw new Exception("Cannot be negative or zero");
    }
  }
}

//Define public property Height
public double Height {
  //Define the getter
  get {
    return this._height;
  }
```

```
    //Define the setter
    set {
      if (value > 0) {
        this._height = value;
      }
      else {
        throw new Exception("Cannot be negative or zero");
      }
    }
  }

  public void displayVolume() {
    Console.WriteLine("Volume: " + (this.Width * this.Length * this.Height));
  }

  public void displayDimensions() {
    Console.WriteLine(this.Width + " x " + this.Length + " x " + this.Height);
  }
}
```

6. Solution

```
double edge;

Console.Write("Enter edge length of a cube: ");
edge = Convert.ToDouble(Console.ReadLine());

Cube cube1 = new(edge);

cube1.displayVolume();
cube1.displayOneSurface();
cube1.displayTotalSurface();

class Cube {
  private double _edge;

  //Define the constructor
  public Cube(double edge) {
    this._edge = edge;
  }

  public void displayVolume() {
    Console.WriteLine("Volume: " + Math.Pow(this._edge, 3));
  }

  public void displayOneSurface() {
    Console.WriteLine("One surface: " + Math.Pow(this._edge, 2));
  }

  public void displayTotalSurface() {
    Console.WriteLine("Total surface: " + 6 * Math.Pow(this._edge, 2));
  }
}
```

7. Solution

```
double edge;

Console.Write("Enter edge length of a cube: ");
edge = Convert.ToDouble(Console.ReadLine());

Cube cube1 = new(edge);

cube1.displayVolume();
cube1.displayOneSurface();
cube1.displayTotalSurface();

class Cube {
  private double _edge;

  //Define the constructor
  public Cube(double edge) {
    //Initialize field (using the corresponding property)
    this.Edge = edge;
  }

 //Define public property Edge
  public double Edge {
    //Define the getter
    get {
      return this._edge;
    }

    //Define the setter
    set {
      if (value > 0) {
        this._edge = value;
      }
      else {
        throw new Exception("Cannot be negative or zero");
      }
    }
  }

  public void displayVolume() {
    Console.WriteLine("Volume: " + Math.Pow(this.Edge, 3));
  }

  public void displayOneSurface() {
    Console.WriteLine("One surface: " + Math.Pow(this.Edge, 2));
  }

  public void displayTotalSurface() {
    Console.WriteLine("Total surface: " + 6 * Math.Pow(this.Edge, 2));
  }
}
```

8. Solution

```
void displayMenu() {
  Console.WriteLine("1. Enter radius");
  Console.WriteLine("2. Display radius");
  Console.WriteLine("3. Display diameter");
  Console.WriteLine("4. Display area");
  Console.WriteLine("5. Display perimeter");
  Console.WriteLine("6. Exit");
}

int choice;
double radius;

Circle circle1 = new();

do {
  displayMenu();
  Console.Write("Enter a choice: ");
  choice = Convert.ToInt32(Console.ReadLine());

  if (choice == 1) {
    Console.Write("Enter radius: ");
    radius = Convert.ToDouble(Console.ReadLine());
    circle1.Radius = radius;
  }
  else if (choice == 2) {
    Console.WriteLine("Radius: " + circle1.Radius);
  }
  else if (choice == 3) {
    Console.WriteLine("Diameter: " + circle1.getDiameter());
  }
  else if (choice == 4) {
    Console.WriteLine("Area: " + circle1.getArea());
  }
  else if (choice == 5) {
    Console.WriteLine("Perimeter: " + circle1.getPerimeter());
  }
} while (choice != 6);

class Circle {
  private double _radius = -1;

  //Define public property Radius
  public double Radius {
    //Define the getter
    get {
      if (this._radius > 0) {
        return this._radius;
      }
      else {
        throw new Exception("Radius is not set");
```

```
      }
    }

    //Define the setter
    set {
      if (value > 0) {
        this._radius = value;
      }
      else {
        throw new Exception("Cannot be negative or zero");
      }
    }
  }

  public double getDiameter() {
    return 2 * this.Radius;
  }

  public double getArea() {
    return 3.14 * Math.Pow(this.Radius, 2);
  }

  public double getPerimeter() {
    return 2 * 3.14 * this.Radius;
  }
}
```

9. Solution

```
Info inf = new();

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

Console.WriteLine("Text: " + inf.UserText);
Console.WriteLine("Spaces: " + inf.getSpacesCount());
Console.WriteLine("Words: " + inf.getWordsCount());
Console.WriteLine("Vowels: " + inf.getVowelsCount());
Console.WriteLine("Total number of letters: " + inf.getLettersCount());

class Info {
  private string _userText;

  //Define public property UserText
  public string UserText {
    //Define the getter
    get {
      return this._userText;
    }

    //Define the setter
    set {
      if (value != "") {
```

```
        this._userText = value;
      }
      else {
        throw new Exception("Cannot be set to empty");
      }
    }
  }

  public int getSpacesCount() {
    int i, count = 0;
    string character;

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

  public int getWordsCount() {
    return this.getSpacesCount() + 1;
  }

  public int getVowelsCount() {
    int i, count = 0;
    string character;

    for (i = 0; i <= this.UserText.Length - 1; i++) {
      character = "" + this.UserText.toLower()[i];
      if ("aeiou".IndexOf(character) > -1) {
        count += 1;
      }
    }
    return count;
  }

  public int getLettersCount() {
    return this.UserText.Length - this.getSpacesCount();
  }
}
```

10. Solution

```
void displayMenu() {
  Console.WriteLine("1. Enter encryption/decryption key");
  Console.WriteLine("2. Encrypt a message");
  Console.WriteLine("3. Decrypt a message");
  Console.WriteLine("4. Exit");
}

string text;
```

```
int choice;

EncryptDecrypt ed = new();

displayMenu();
Console.Write("Enter a choice: ");
choice = Convert.ToInt32(Console.ReadLine());
while (choice != 4) {
  if (choice == 1) {
    Console.Write("Enter encryption/decryption key: ");
    ed.EncrDecrKey = Convert.ToInt32(Console.ReadLine());
  }
  else if (choice == 2) {
    Console.Write("Enter message to encrypt: ");
    text = Console.ReadLine();
    Console.WriteLine("Encrypted message: " + ed.encrypt(text));
  }
  else if (choice == 3) {
    Console.Write("Enter message to decrypt: ");
    text = Console.ReadLine();
    Console.WriteLine("Decrypted message: " + ed.decrypt(text));
  }

  displayMenu();
  Console.Write("Enter a choice: ");
  choice = Convert.ToInt32(Console.ReadLine());
}

class EncryptDecrypt {
  const string alphabet = " abcdefghijklmnopqrstuvwxyz";  //space is a valid character!
  private int _encrDecrKey = -1;

  //Define the propert EncrDecrKey
  public int EncrDecrKey {
    //Define the getter
    get {
      if (this._encrDecrKey != -1) {
        return this._encrDecrKey;
      }
      else {
        throw new Exception("Key is not set");
      }
    }

    //Define the setter
    set {
      if (value >= 1 && value <= 26) {
        this._encrDecrKey = value;
      }
      else {
        throw new Exception("Must be between 1 and 26");
      }
```

```
    }
  }

  public string encrypt(string message) {
    string character, returnValue = "";
    char newLetter;
    int i, index, newIndex;

    for (i = 0; i <= message.Length - 1; i++) {
      character = "" + message[i];
      index = alphabet.IndexOf(character);
      newIndex = index + this.EncrDecrKey;
      if (newIndex >= 27) {
        newIndex -= 27;
      }
      newLetter = alphabet[newIndex];
      returnValue += newLetter;
    }
    return returnValue;
  }

  public string decrypt(string encMessage) {
    string character, returnValue = "";
    char newLetter;
    int i, index, newIndex;

    for (i = 0; i <= encMessage.Length - 1; i++) {
      character = "" + encMessage[i];
      index = alphabet.IndexOf(character);
      newIndex = index - this.EncrDecrKey;
      if (newIndex < 0) {
        newIndex += 27;
      }
      newLetter = alphabet[newIndex];
      returnValue += newLetter;
    }
    return returnValue;
  }
}
```

11. Solution

```
Car car1 = new(4, "Red", 5, 2, 1.5);
car1.bootCapacity = 300;
car1.startEngine();
car1.turnWindshieldWipersOn();
car1.stopEngine();

Car car2 = new(4, "Green", 4.5, 2.2, 1.4);
car2.bootCapacity = 400;
car2.startEngine();
car2.turnWindshieldWipersOn();
```

```
car2.stopEngine();

Motorcycle motorcycle1 = new(2, "Blue", 2, 0.9, 1.3);
motorcycle1.hasLuggage = true;
motorcycle1.startEngine();
motorcycle1.doAWheelie();
motorcycle1.stopEngine();

class Vehicle {
  public int numberOfWheels;
  public string color;
  public double length, width, height;

  //Define the constructor
  public Vehicle(int numberOfWheels, string color, double length, double width, double height) {
    this.numberOfWheels = numberOfWheels;
    this.color = color;
    this.length = length;
    this.width = width;
    this.height = height;
  }

  public void startEngine() {
    Console.WriteLine("The engine started");
  }

  public void stopEngine() {
    Console.WriteLine("The engine stopped");
  }
}

class Car : Vehicle {
  public int bootCapacity;

  //Define the constructor
  public Car(int numberOfWheels, string color, double length, double width, double height)
            : base(numberOfWheels, color, length, width, height) {
    this.bootCapacity = 0;
  }

  public void turnWindshieldWipersOn() {
    Console.WriteLine("The windshield wipers have been turned on!");
  }
}

class Motorcycle : Vehicle {
  public bool hasLuggage;

  //Define the constructor
  public Motorcycle(int numberOfWheels, string color, double length, double width, double height)
                   : base(numberOfWheels, color, length, width, height) {
    this.hasLuggage = false;
  }
```

```
  public void doAWheelie() {
    Console.WriteLine("I am doing a wheelie!!!");
  }
}
```

12. Solution

```
Teacher teacher1 = new("Mr. John Scott", 43, 35000);
Teacher teacher2 = new("Mrs. Ann Carter", 5, 32000);

Student student1 = new("Mark Nelson", 14, "A");
Student student2 = new("Mary Morgan", 13, "B");

teacher1.displayValues();
teacher2.displayValues();
student1.displayValues();
student2.displayValues();

class SchoolMember {
  private string _name;
  private int _age;

  public SchoolMember(string name, int age) {
    this.setName(name);
    this.setAge(age);
    Console.WriteLine("A school member was initialized");
  }

  public string getName() {
    return this._name;
  }

  void setName(string value) {
    if (value != "") {
      this._name = value;
    }
    else {
      throw new Exception("Name cannot be empty");
    }
  }

  public int getAge() {
    return this._age;
  }

  public void setAge(int value) {
    if (value > 0) {
      this._age = value;
    }
    else {
      throw new Exception("Age cannot be negative or zero");
    }
```

```
  }
}

class Teacher : SchoolMember {
  private double _salary;

  public Teacher(string name, int age, double salary) : base(name, age) {
    this.setSalary(salary);
    Console.WriteLine("A teacher was initialized");
  }

  public void displayValues() {
    Console.WriteLine("Name: " + this.getName());
    Console.WriteLine("Age: " + this.getAge());
    Console.WriteLine("Salary: " + this.getSalary());
  }

  public double getSalary() {
    return this._salary;
  }

  public void setSalary(double value) {
    if (value >= 0) {
      this._salary = value;
    }
    else {
      throw new Exception("Salary cannot be negative");
    }
  }
}

class Student : SchoolMember {
  private string _finalGrade;

  public Student(string name, int age, string finalGrade) : base(name, age) {
    this.setFinalGrade(finalGrade);
    Console.WriteLine("A student was initialized");
  }

  public void displayValues() {
    Console.WriteLine("Name: " + this.getName());
    Console.WriteLine("Age: " + this.getAge());
    Console.WriteLine("Salary: " + this.getFinalGrade());
  }

  public string getFinalGrade() {
    return this._finalGrade;
  }

  public void setFinalGrade(string value) {
    if (value == "A" || value == "B" || value == "C" || value == "D" || value == "E" || value == "F") {
      this._finalGrade = value;
    }
```

```
    else {
      throw new Exception("Final grade must be in the range of 'A' to 'F'");
    }
  }
}
```

13. Solution

```
Teacher teacher1 = new("Mr. John Scott", 43, 35000);
Teacher teacher2 = new("Mrs. Ann Carter", 5, 32000);

Student student1 = new("Mark Nelson", 14, "A");
Student student2 = new("Mary Morgan", 13, "B");

teacher1.displayValues();
teacher2.displayValues();
student1.displayValues();
student2.displayValues();

class SchoolMember {
  private string _name;
  private int _age;

  public SchoolMember(string name, int age) {
    this.Name = name;
    this.Age = age;
    Console.WriteLine("A school member was initialized");
  }

  public string Name {
    get {
      return this._name;
    }

    set {
      if (value != "" ) {
        this._name = value;
      }
      else {
        throw new Exception("Name cannot be empty");
      }
    }
  }

  public int Age {
    get {
      return this._age;
    }

    set {
      if (value > 0) {
        this._age = value;
      }
```

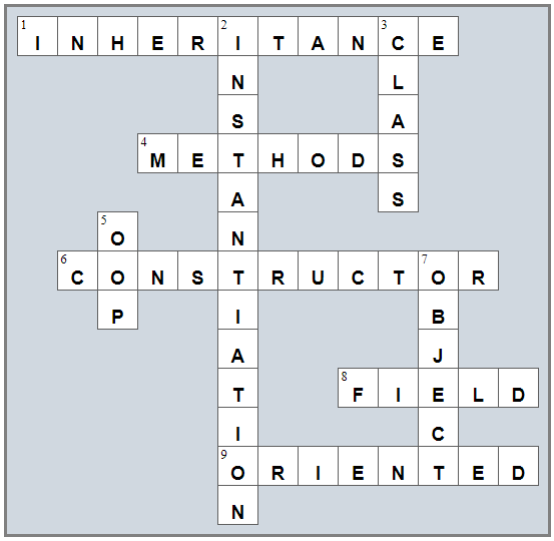
```
      else {
        throw new Exception("Age cannot be negative or zero");
      }
    }
  }
}

class Teacher : SchoolMember {
  private double _salary;

  public Teacher(string name, int age, double salary) : base(name, age) {
    this.Salary = salary;
    Console.WriteLine("A teacher was initialized");
  }

  public void displayValues() {
    Console.WriteLine("Name: " + this.Name);
    Console.WriteLine("Age: " + this.Age);
    Console.WriteLine("Salary: " + this.Salary);
  }

  public double Salary {
    get {
      return this._salary;
    }

    set {
      if (value >= 0) {
        this._salary = value;
      }
      else {
        throw new Exception("Salary cannot be negative");
      }
    }
  }
}

class Student : SchoolMember {
  private string _finalGrade;

  public Student(string name, int age, string finalGrade) : base(name, age) {
    this.FinalGrade = finalGrade;
    Console.WriteLine("A student was initialized");
  }

  public void displayValues() {
    Console.WriteLine("Name: " + this.Name);
    Console.WriteLine("Age: " + this.Age);
    Console.WriteLine("Salary: " + this.FinalGrade);
  }

  public string FinalGrade {
    get {
```

```
      return this._finalGrade;
    }

    set {
      if (value == "A" || value == "B" || value == "C" || value == "D" || value == "E" || value == "F") {
        this._finalGrade = value;
      }
      else {
        throw new Exception("Final grade must be in the range of 'A' to 'F'");
      }
    }
  }
}
```

Review in “Object Oriented Programming”

Review Crossword Puzzle

1.



Chapter 40

40.8 Review Questions: True/False

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

40.9 Review Exercises

1. Solution

```
const string PATH = "c:/temp/";

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

StreamWriter f = File.CreateText(PATH + "days_of_week.txt");
foreach (var d in days) {
  f.WriteLine(d);
}
f.Close();
```

2. Solution

```
const string PATH = "c:/temp/";

int i;
string[] days = new string[7];

StreamReader f = File.OpenText(PATH + "days_of_week.txt");
for (i = 0; i <= 6; i++) {
  days[i] = f.ReadLine();
}
f.Close();

for (i = 6; i >= 0; i--) {
  Console.WriteLine(days[i]);
}
```

3. Solution

```
const string PATH = "c:/temp/";

StreamWriter f = File.AppendText(PATH + "days_of_week.txt");
f.WriteLine("*** End of File ***");
f.Close();
```

4. Solution

```
const string PATH = "c:/temp/";

int i;
Random rnd = new();

StreamWriter f = File.CreateText(PATH + "randoms.txt");
for (i = 0; i <= 49; i++) {
  f.WriteLine(rnd.Next(1, 101));
}
f.Close();
```

5. Solution

```
const string PATH = "c:/temp/";

int i;
StreamWriter f;
Random rnd = new();

for (i = 1; i <= 10; i++) {
  f = File.CreateText(PATH + "file" + i + ".txt");
  f.Write(rnd.Next(100, 10000));
  f.Close();
}
```

6. Solution

```
const string PATH = "c:/temp/";

int i, j;
StreamWriter f = File.CreateText(PATH + "multiplication_table.txt");

for (i = 1; i <= 10; i++) {
  for (j = 1; j <= 4; j++) {
    f.WriteLine(i + " x " + j + " = " + (i * j));
  }
}
f.Close();
```

7. Solution

```
const string PATH = "c:/temp/";

StreamReader f = File.OpenText(PATH + "a_file.txt");

while (!f.EndOfStream) {
  Console.WriteLine(f.ReadLine().Length);
}
f.Close();
```

8. Solution

First approach

```
const string PATH = "c:/temp/";

int i;
string line;
StreamReader f = File.OpenText(PATH + "a_file.txt");

i = 1;
while (!f.EndOfStream) {
  line = f.ReadLine();
  foreach (var character in line) {
    if (",.!".IndexOf(character) > -1) {
      Console.WriteLine("There is a punctuation mark on line No " + i);
      break;
    }
  }
  i++;
}

f.Close();
```

Second approach

```
const string PATH = "c:/temp/";

int i;
string line;
StreamReader f = File.OpenText(PATH + "a_file.txt");

i = 1;
while (!f.EndOfStream) {
  line = f.ReadLine();
  if (line.IndexOf(",") > -1 || line.IndexOf(".") > -1 || line.IndexOf("!") > -1) {
    Console.WriteLine("There is a punctuation mark on line No " + i);
  }
  i++;
}

f.Close();
```

Chapter 41

41.2 Review Exercises

1. Solution

```
const string PATH = "c:/temp/";

int i, total, count, number;

StreamReader fin = File.OpenText(PATH + "f_data41.2-1.txt");

string values = fin.ReadLine();
fin.Close();

total = 0;
count = 0;
for (i = 0; i < 10; i++) {
  number = Convert.ToInt32(values.Substring(i * 3, 2));
  if (number > 50) {
    total += number;
    count += 1;
  }
}

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

2. Solution

```
const string PATH = "c:/temp/";

int i, total, count, number;

StreamReader fin = File.OpenText(PATH + "f_data41.2-2.txt");
string values = fin.ReadLine();
fin.Close();

total = 0;
count = 0;
i = 0;
while (i < values.Length / 4) {
  number = Convert.ToInt32(values.Substring(i * 4, 3));
  if (number >= 300 && number <= 500) {
    total += number;
    count += 1;
  }
  i++;
}

if (count > 0) {
```

```
  Console.WriteLine(total / (double)count);
}
```

3. Solution

```
const string PATH = "c:/temp/";

string maxName, minName, name, line;
int maximum, minimum, grade, commaPosition;

StreamReader fin = File.OpenText(PATH + "f_data41.2-3.txt");

//Read the first line
line = fin.ReadLine();

commaPosition = line.IndexOf(',');
grade = Convert.ToInt32(line.Substring(0, commaPosition));
name = line.Substring(commaPosition + 1);

maximum = minimum = grade;
maxName = minName = name;

//Read the rest of the lines
while (!fin.EndOfStream) {
  line = fin.ReadLine();

  commaPosition = line.IndexOf(',');
  grade = Convert.ToInt32(line.Substring(0, commaPosition));
  name = line.Substring(commaPosition + 1);

  if (grade > maximum) {
    maximum = grade;
    maxName = name;
  }
  if (grade < minimum) {
    minimum = grade;
    minName = name;
  }
}

fin.Close();

Console.WriteLine(maxName);
Console.WriteLine(minName);
```

4. Solution

```
const string PATH = "c:/temp/";

string description, maximumDescription, keyword, stringInfo1, stringInfo2, line;
double width, length, height, volume, total;
double maximum;

Console.Write("Enter keyword to search: ");
```

```
keyword = Console.ReadLine();

StreamReader fin = File.OpenText(PATH + "f_data41.2-4.txt");

maximum = total = 0;
maximumDescription = stringInfo1 = stringInfo2 = "";
while (!fin.EndOfStream) {
  line = fin.ReadLine();
  width = Convert.ToDouble(line.Substring(0, 5));
  length = Convert.ToDouble(line.Substring(6, 5));
  height = Convert.ToDouble(line.Substring(12, 5));
  description = line.Substring(18);

  if (description.ToLower().IndexOf(keyword) != -1) {
    stringInfo1 += description + " - Dimensions: " + width + " x " + length + " x " + height + "\n";

  }
  volume = width * length * height / 1728;
  stringInfo2 += description + " - Volume = " + volume + " cubic feet\n";

  total += volume;

  if (volume > maximum) {
    maximum = volume;
    maximumDescription = description;
  }
}
fin.Close();

if (stringInfo1 != "") {
  Console.WriteLine("Keyword ' " + keyword + "' found!");
  Console.WriteLine(stringInfo1);
}

Console.WriteLine("Volume of each item:");
Console.WriteLine(stringInfo2);

Console.WriteLine("Total volume: " + total);
Console.WriteLine("Greatest box: " + maximumDescription);
```

5. Solution

First approach

```
string filename1, filename2, content;

Console.Write("Enter filename No 1: ");
filename1 = Console.ReadLine();

if (filename1.Substring(filename1.Length - 4) != ".txt") {
  Console.WriteLine("Wrong filename");
}
else {
  Console.Write("Enter filename No 2: ");
  filename2 = Console.ReadLine();
```

```
  if (filename2.Substring(filename1.Length - 4) != ".txt") {
    Console.WriteLine("Wrong filename");
  }
  else {
    StreamReader fin;

    fin = File.OpenText(filename2);
    content = fin.ReadToEnd();
    fin.Close();

    fin = File.OpenText(filename1);
    content += fin.ReadToEnd();    //Concatenation
    fin.Close();

    StreamWriter fout = File.CreateText("final.txt");
    fout.Write(content);
    fout.Close();
  }
}
```

Second approach

```
string filename1, filename2;

Console.Write("Enter filename No 1: ");
filename1 = Console.ReadLine();

if (filename1.Substring(filename1.Length - 4) != ".txt") {
  Console.WriteLine("Wrong filename");
}
else {
  Console.Write("Enter filename No 2: ");
  filename2 = Console.ReadLine();
  if (filename2.Substring(filename1.Length - 4) != ".txt") {
    Console.WriteLine("Wrong filename");
  }
  else {
    StreamReader fin1 = File.OpenText(filename1);
    StreamReader fin2 = File.OpenText(filename2);
    StreamWriter fout = File.CreateText("final.txt");

    fout.Write(fin2.ReadToEnd() + fin1.ReadToEnd());

    fout.Close();
    fin2.Close();
    fin1.Close();
  }
}
```

6. Solution

```
const string PATH = "c:/temp/";
const int ELEMENTS = 15;

int i, m, n;
```

```
double[] numbers = new double[ELEMENTS];
double temp;

StreamReader fin = File.OpenText(PATH + "f_data41.2-6.txt");
for (i = 0; i < ELEMENTS; i++) {
  numbers[i] = Convert.ToDouble(fin.ReadLine());
}
fin.Close();

for (m = 1; m <= ELEMENTS - 1; m++) {
  for (n = ELEMENTS - 1; n >= m; n--) {
    if (numbers[n] < numbers[n - 1]) {
      temp = numbers[n];
      numbers[n] = numbers[n - 1];
      numbers[n - 1] = temp;
    }
  }
}

StreamWriter fout = File.AppendText(PATH + "f_data41.2-6.txt");
fout.WriteLine("\n********* Sorted numbers *************");
foreach (var number in numbers) {
  fout.WriteLine(number);
}
fout.Close();
```

7. Solution

```
const string PATH = "c:/temp/";
const int NUMBER_OF_CITIES = 8;

int i;
bool onCityLine;
double total, average, maximum;

string[] cities = new string[NUMBER_OF_CITIES];
double[] temperatures = new double[NUMBER_OF_CITIES];

StreamReader fin = File.OpenText(PATH + "f_data41.2-7.txt");

//Split read values into two arrays (cities and temperatures)
i = 0;
onCityLine = true;
while (!fin.EndOfStream) {
  if (onCityLine) {
    cities[i] = fin.ReadLine();
  }
  else {
    temperatures[i++] = Convert.ToDouble(fin.ReadLine());
  }
  onCityLine = !onCityLine;  //true becomes false, and false becomes true
}
```

```
fin.Close();

total = 0;
for (i = 0; i < NUMBER_OF_CITIES; i++) {
  total += temperatures[i];
}

average = total / NUMBER_OF_CITIES;
Console.WriteLine(average);

maximum = temperatures.Max();
Console.WriteLine("Highest temperature: " + maximum);
for (i = 0; i < NUMBER_OF_CITIES; i++) {
  if (temperatures[i] == maximum) {
    Console.WriteLine(cities[i]);
  }
}
```

8. Solution

```
const string PATH = "c:/temp/";

string abbreviate(string word) {
  if (word.Length > 10) {
    return "" + word[0] + (word.Length - 2) + word[word.Length - 1];
  }
  else {
    return word;
  }
}

string line, word;
int spaceIndex;
StreamReader fin = File.OpenText(PATH + "f_data41.2-8.txt");

while (!fin.EndOfStream) {
  line = fin.ReadLine();

  spaceIndex = line.IndexOf(" ");  //Find the first space
  while (spaceIndex > -1) {
    word = line.Substring(0, spaceIndex);   //Get the word and
    line = line.Substring(spaceIndex + 1);  //remove the word from line

    Console.Write(abbreviate(word) + " ");

    spaceIndex = line.IndexOf(" ");
  }

  //Display the last word remained in the string line
  Console.WriteLine(abbreviate(line));
}

fin.Close();
```

9. Solution

```
const string PATH = "c:/temp/";

string pigLatinTranslator(string word) {
  int i, firstVowelIndex;
  string pigLatinWord, vowels = "aeiou";

  if (vowels.IndexOf(word[0]) != -1) { //If first character is vowel
    pigLatinWord = word + "way";
  }
  else {
    //Find the index of the first vowel
    firstVowelIndex = -1;
    for (i = 0; i <= word.Length - 1; i++) {
      if (vowels.IndexOf(word[i]) != -1) {
        firstVowelIndex = i;
        break;
      }
    }

    //If at least one vowel found
    if (firstVowelIndex != -1) {
      //Move the consonants to the end
      word = word.Substring(firstVowelIndex) + word.Substring(0, firstVowelIndex);
    }
    pigLatinWord = word + "ay";
  }
  return pigLatinWord;
}

string line, word;
int spaceIndex;
StreamReader fin = File.OpenText(PATH + "f_data41.2-9.txt");
StreamWriter fout = File.CreateText(PATH + "pig_latin_translation.txt");

while (!fin.EndOfStream) {
  line = fin.ReadLine();

  spaceIndex = line.IndexOf(" ");  //Find the first space
  while (spaceIndex > -1) {
    word = line.Substring(0, spaceIndex);   //Get the word and
    line = line.Substring(spaceIndex + 1);  //remove the word from line

    fout.Write(pigLatinTranslator(word) + " ");

    spaceIndex = line.IndexOf(" ");
  }
  //Write the last word remained in the string line
  fout.WriteLine(pigLatinTranslator(line));
}

fin.Close();
```

```
fout.Close();
```

10. Solution

```
const string PATH = "c:/temp/";

const string X = " ABCDEFGHIJKLMNOPQRSTUVWXYZ";   //The space character remains as is
const string Y = " JKWCTAMEDXSLFBYUNGRZOIQVHP";

string initialMessage, encryptedMessage;
int i;

Console.Write("Enter a message to encrypt: ");
initialMessage = Console.ReadLine().ToUpper();

encryptedMessage = "";
foreach (var letter in initialMessage) {
  //Search for letter in const X
  for (i = 0; i < 27; i++) {
    if (letter == X[i]) {
      //Create encrypted message using letters from const Y
      encryptedMessage += Y[i];
      break;
    }
  }
}

StreamWriter fout = File.CreateText(PATH + "encrypted.txt");
fout.Write(encryptedMessage);
fout.Close();
```

11. Solution

```
const string PATH = "c:/temp/";

const string X = " ABCDEFGHIJKLMNOPQRSTUVWXYZ";   //The space character remains as is
const string Y = " JKWCTAMEDXSLFBYUNGRZOIQVHP";

string initialMessage, encryptedMessage;
int i;

StreamReader fin = File.OpenText(PATH + "encrypted.txt");
encryptedMessage = fin.ReadLine();
fin.Close();

initialMessage = "";
foreach (var letter in encryptedMessage) {
  //Search for letter in const Y
  for (i = 0; i < 27; i++) {
    if (letter == Y[i]) {
      //Create decrypted message using letters from const X
      initialMessage += X[i];
      break;
    }
```

```
  }
}

StreamWriter fout = File.CreateText(PATH + "decrypted.txt");
fout.Write(initialMessage);
fout.Close();
```

12. Solution

First approach

```
void copyFile(string source, string destination) {
  StreamReader fin = File.OpenText(source);
  string x = fin.ReadToEnd();
  fin.Close();

  StreamWriter fout = File.CreateText(destination);
  fout.Write(x);
  fout.Close();
}
```

Second approach

```
void copyFile(string source, string destination) {
  StreamReader fin = File.OpenText(source);
  StreamWriter fout = File.CreateText(destination);

  fout.Write(fin.ReadToEnd());

  fin.Close();
  fout.Close();
}
```

13. Solution

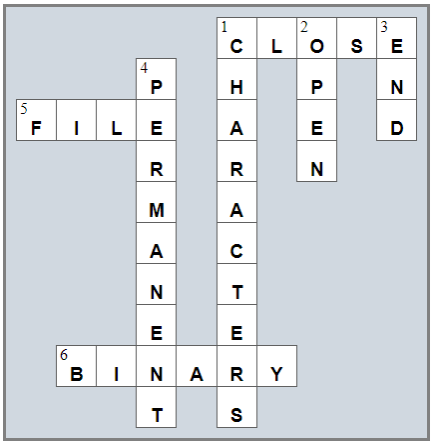
```
Triangle tr = new();

tr.displayLengths();
tr.displayArea();
tr.displayPerimeter();

class Triangle {
  const string PATH = "c:/temp/";

  private double _sideA, _sideB, _sideC;

  //Define the constructor
  public Triangle() {
    StreamReader fin = File.OpenText(PATH + "f_data41.2-13.txt");
    this._sideA = Convert.ToDouble(fin.ReadLine());
    this._sideB = Convert.ToDouble(fin.ReadLine());
    this._sideC = Convert.ToDouble(fin.ReadLine());
    fin.Close();
  }

  public bool canBeTriangle() {
```

```
    if (this._sideA > 0 && this._sideB > 0 && this._sideC > 0 &&
        this._sideA + this._sideB > this._sideC &&
        this._sideB + this._sideC > this._sideA &&
        this._sideC + this._sideA > this._sideB) {
      return true;
    }
    else {
      return false;
    }
  }

  public void displayLengths() {
    Console.WriteLine("Side A: " + this._sideA);
    Console.WriteLine("Side B: " + this._sideB);
    Console.WriteLine("Side C: " + this._sideC);
    if (this.canBeTriangle()) {
      Console.WriteLine("Can be lengths of the three sides of a triangle!");
    }
    else {
      Console.WriteLine("Cannot be lengths of the three sides of a triangle!");
    }
  }

  public void displayArea() {
    double s, area;

    if (this.canBeTriangle()) {
      s = (this._sideA + this._sideB + this._sideC) / 2;
      area = Math.Sqrt(s * (s - this._sideA) * (s - this._sideB) * (s - this._sideC));
      Console.WriteLine("Area: " + area);
    }
  }

  public void displayPerimeter() {
    double perimeter;

    if (this.canBeTriangle()) {
      perimeter = this._sideA + this._sideB + this._sideC;
      Console.WriteLine("Perimeter: " + perimeter);
    }
  }
}
```

Review in “Files”

Review Crossword Puzzle

1.



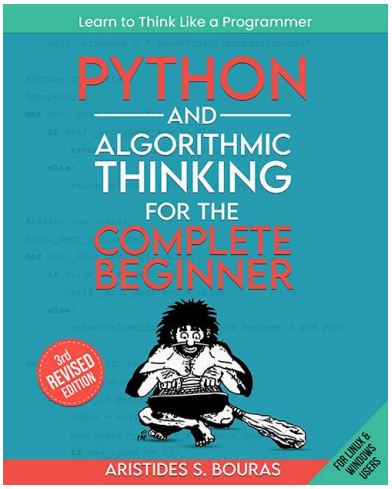
Some Final Words from the Author

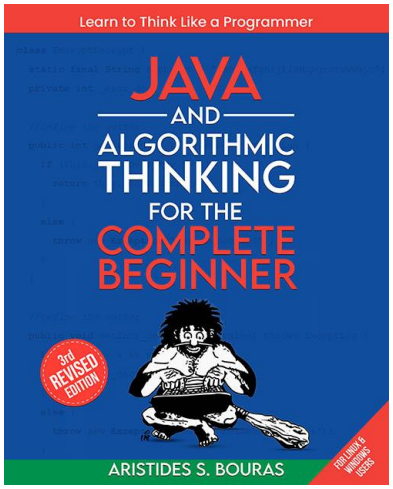
I hope you thoroughly enjoyed reading this book. I made every possible effort to ensure it is beneficial and comprehensible, even for people who may have no prior experience in programming.

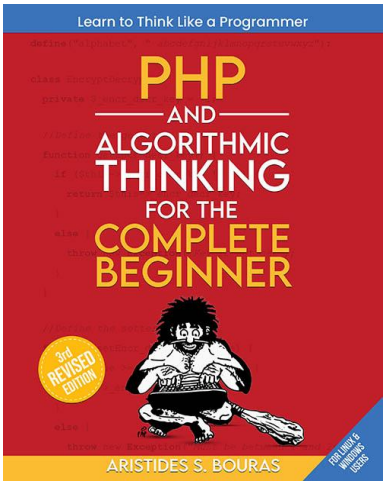
If you found this book valuable, please consider visiting the web store where you purchased it, as well as goodreads.com, to show your appreciation by writing a positive review and awarding as many stars as you think appropriate. By doing so, you will motivate me to keep writing and, of course, you'll be assisting other readers in discovering my work.

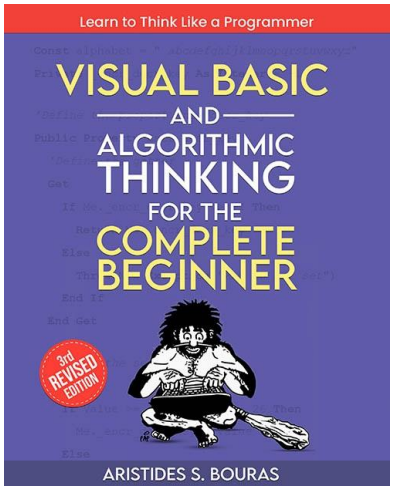
And always remember: Learning is a lifelong, continuous process that begins at birth and extends throughout your lifetime!

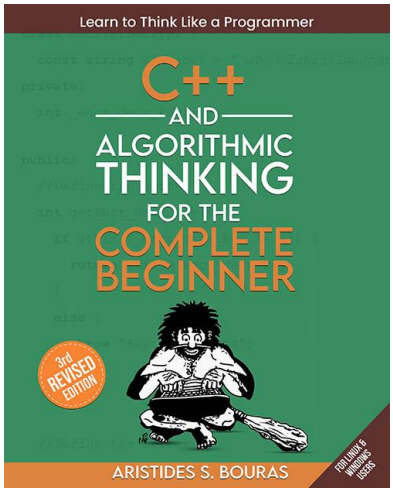
Some of my Books

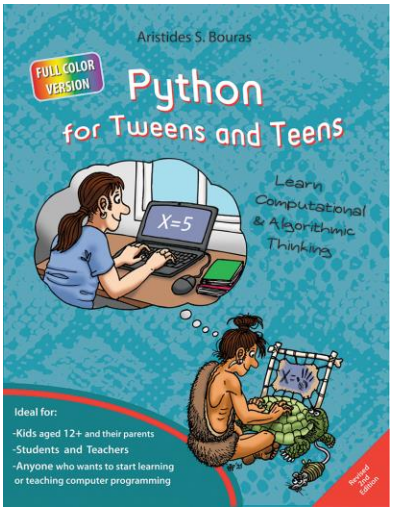


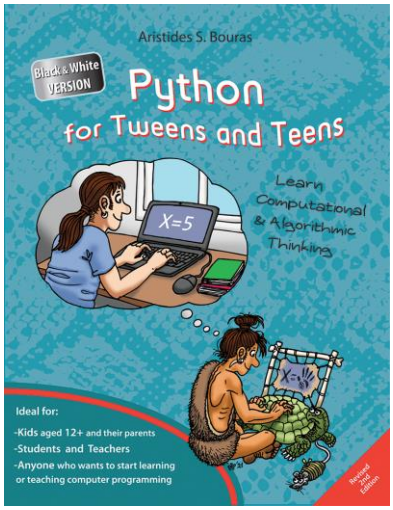


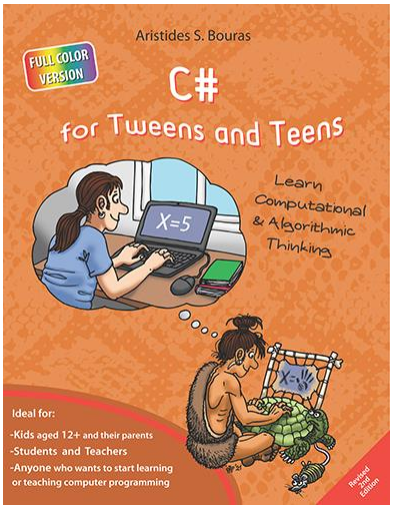


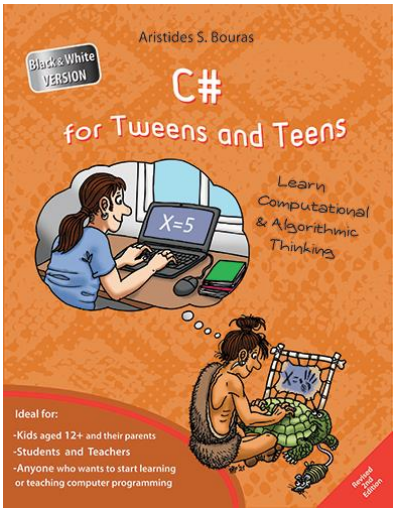


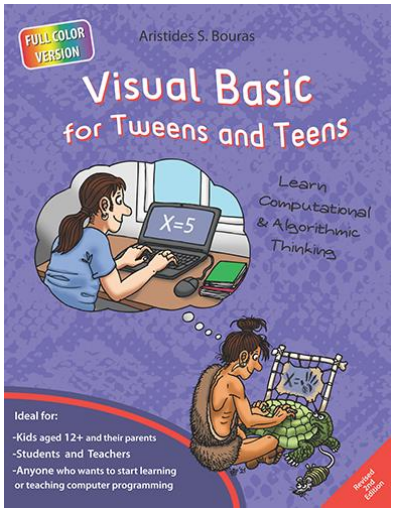


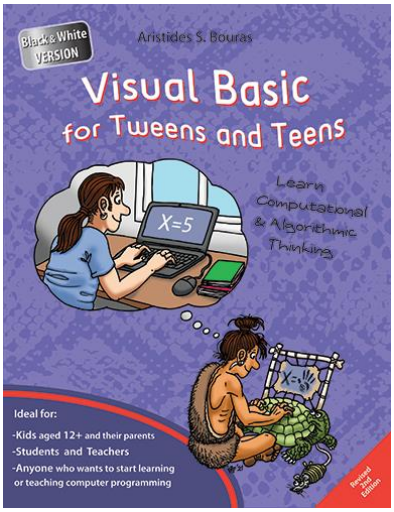












For more information about my books visit my website:

https://www.bouraspage.com