VISUAL BASIC
AND ALGORITHMIC THINKING FOR THE COMPLETE BEGINNER
Second Edition

The Answers

Aristides S. Bouras

RCode: 221008

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 "VISUAL BASIC AND ALGORITHMIC THINKING FOR THE COMPLETE BEGINNER – Second Edition". Every effort has been taken to make this book compatible with all releases of Visual Basic, 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

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

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

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

Chapter 1

1.7 Review Questions: True/False

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

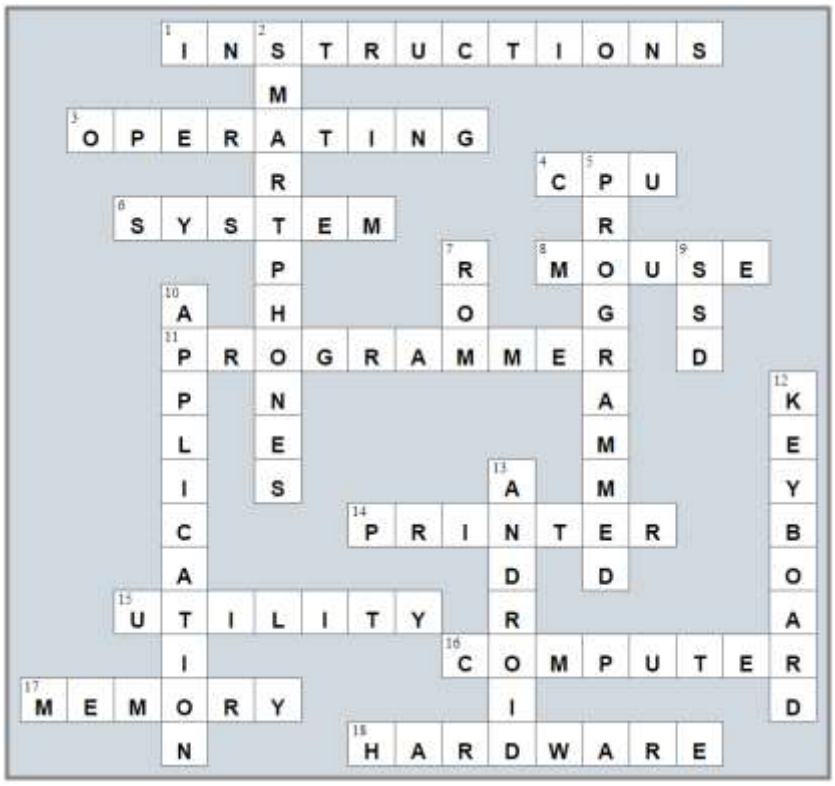
1.8 Review Questions: Multiple Choice

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

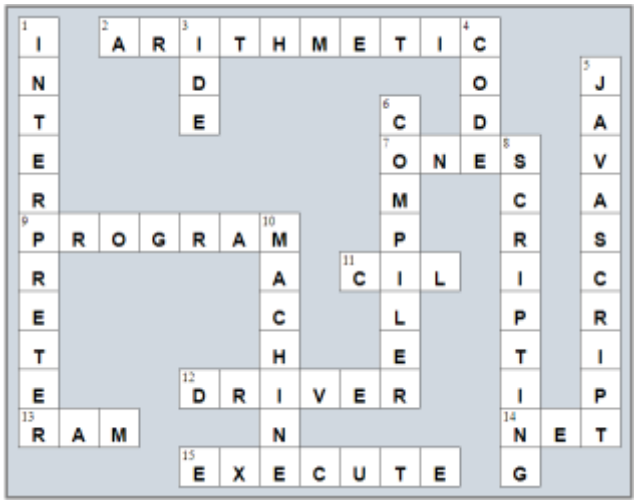
Review in “Introductory Knowledge”

Review Crossword Puzzles

1.



2.



3.

Chapter 4

4.16 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

4.17 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. a
7. d

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	`Dim name As String = "Mark"`
My address	String	`Dim address As String` `address = "254 Lookout Rd. Wilson, NY 27893"`
The average daily temperature	Float	`Dim average As Double = 70.3`
A telephone number	String	`Dim phone_number As String = "1-891-764-2410"`
My Social Security Number (SSN)	String	`Dim ssn As String = "123-45-6789"`
The speed of a car	Float	`Dim speed As Double = 90.5`
The number of children in a family	Integer	`Dim children As Integer = 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. a
2. c
3. b
4. b

Chapter 7

7.6 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

7.7 Review Questions: Multiple Choice

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

7.8 Review Exercises

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

Chapter 8

8.2 Review Questions: True/False

1. False
2. True
3. False
4. False

8.3 Review Exercises

1. Solution

For the input value of 3

Step	Statement	a	b	c	d
1	`a = Console.ReadLine()`	**3**	?	?	?
2	`a = (a + 1) * (a + 1) + 6 / 3 * 2 + 20`	**40**	?	?	?
3	`b = a Mod 13`	40	**1**	?	?
4	`c = b Mod 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 = Console.ReadLine()`	**4**	?	?	?
2	`a = (a + 1) * (a + 1) + 6 / 3 * 2 + 20`	**49**	?	?	?
3	`b = a Mod 13`	49	**10**	?	?
4	`c = b Mod 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 = Console.ReadLine()`	**1**	?	?	?
2	`a = (a + 1) * (a + 1) + 6 / 3 * 2 + 20`	**28**	?	?	?
3	`b = a Mod 13`	28	**2**	?	?
4	`c = b Mod 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			

2. Solution

For the input values of 8, 4

Step	Statement	a	b	c	d	e
1	`a = Console.ReadLine()`	**8.0**	?	?	?	?
2	`b = Console.ReadLine()`	8.0	**4.0**	?	?	?

3	`c = a + b`	8.0	4.0	**12.0**	?	?
4	`d = 1 + a / b * c + 2`	8.0	4.0	12.0	**27.0**	?
5	`e = c + d`	8.0	4.0	12.0	27.0	**39.0**
6	`c += d + e`	8.0	4.0	**78.0**	27.0	39.0
7	`e -= 1`	8.0	4.0	78.0	27.0	**38.0**
8	`d -= c + d Mod c`	8.0	4.0	78.0	**–78.0**	38.0
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 = Console.ReadLine()`	**4.0**	?	?	?	?
2	`b = Console.ReadLine()`	4.0	**4.0**	?	?	?
3	`c = a + b`	4.0	4.0	**8.0**	?	?
4	`d = 1 + a / b * c + 2`	4.0	4.0	8.0	**11.0**	?
5	`e = c + d`	4.0	4.0	8.0	11.0	**19.0**
6	`c += d + e`	4.0	4.0	**38.0**	11.0	19.0
7	`e -= 1`	4.0	4.0	38.0	11.0	**18.0**
8	`d -= c + d Mod c`	4.0	4.0	38.0	**–38.0**	18.0
9	`Console.WriteLine(c & ", " & d & ", " & e)`	It displays: 38, –38, 18				

Chapter 9

9.4 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 = Console.ReadLine()`	**5**	?	?	?	?
2	`b = 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 -= 1`	5	5	37	17	**–8**
8	`d += e + c Mod 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 = Console.ReadLine()`	**4**	?	?	?	?
2	`b = 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 -= 1`	4	2	31	19	**–14**
8	`d += e + c Mod 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 = Console.ReadLine()`	?	**5**	?
2	`c = 5`	?	5	**5**
3	`c = c * b`	?	5	**25**
4	`a = 3 * c Mod 10`	**5**	5	25

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

For the input value of 4

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

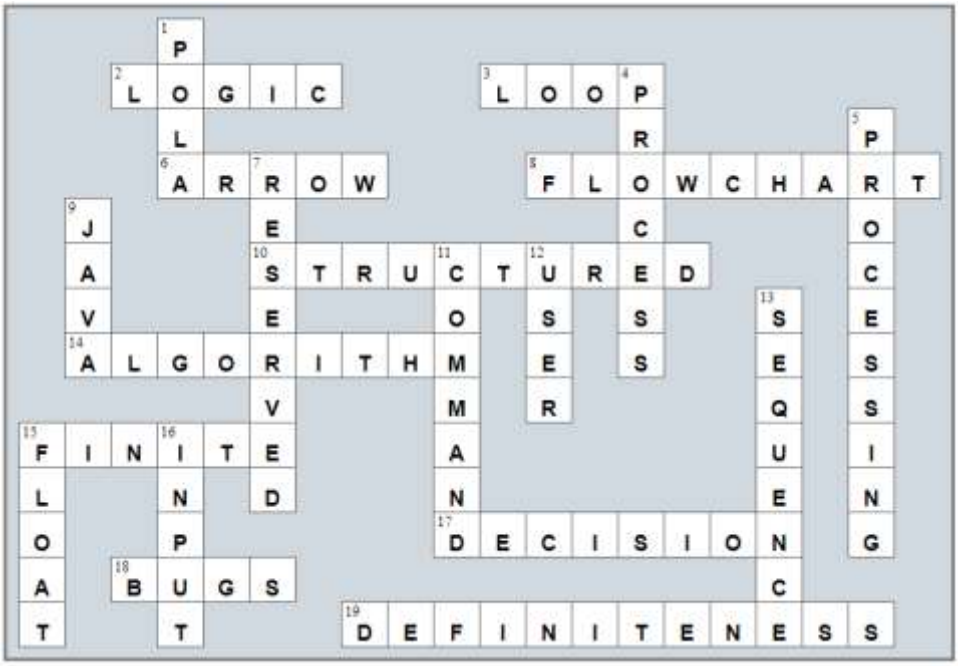
For the input value of 15

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

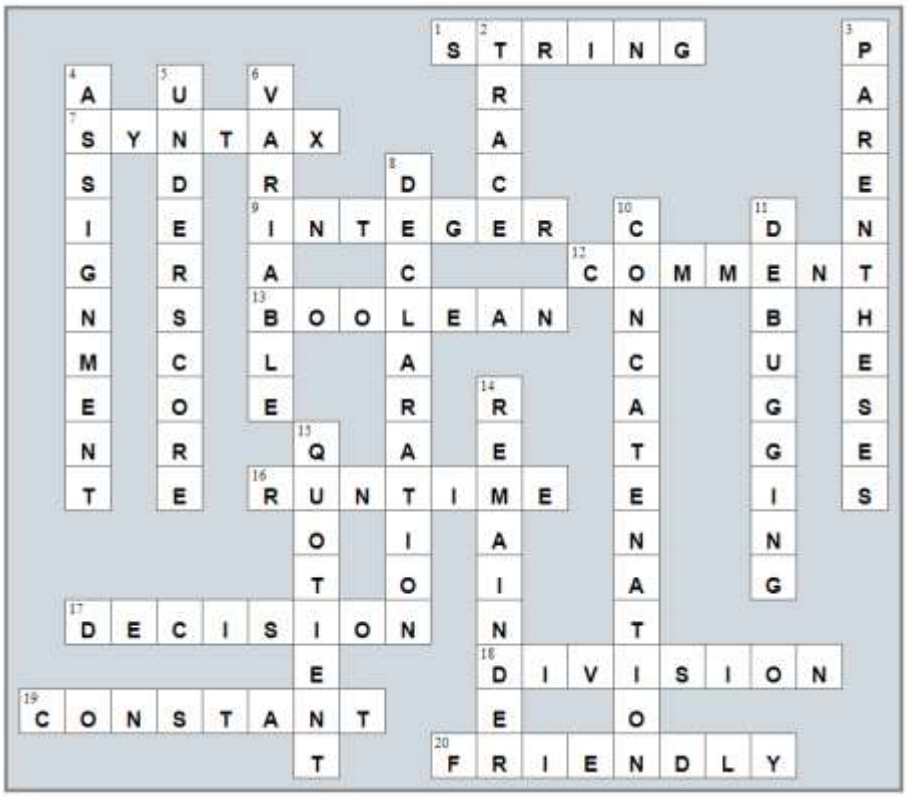
Review in “Getting Started with Visual Basic”

Review Crossword Puzzles

1.



2.



Chapter 10

10.2 Review Exercises

1. Solution

```
Sub Main(args As String())
  Dim b, h, area As Double

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

  area = 0.5 * b * h

  Console.WriteLine(area)
End Sub
```

2. Solution

```
Sub Main(args As String())
  Dim angle1, angle2, angle3 As Double

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

  angle3 = 180 - angle1 - angle2

  Console.WriteLine(angle3)
End Sub
```

3. Solution

```
Sub Main(args As String())
  Dim g1, g2, g3, g4 As Integer
  Dim average As Double

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

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

  Console.WriteLine(average)
End Sub
```

4. Solution

```
Const PI = 3.14159

Sub Main(args As String())
  Dim r, perimeter As Double

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

  perimeter = 2 * PI * r

  Console.WriteLine(perimeter)
End Sub
```

5. Solution

```
Const PI = 3.14159

Sub Main(args As String())
  Dim d, radius, volume As Double

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

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

  Console.WriteLine(volume)
End Sub
```

6. Solution

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

7. Solution

```
Const PI = 3.14159

Sub Main(args As String())
  Dim d, radius, perimeter, area, volume As Double

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

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

  Console.WriteLine(radius & " " & perimeter & " " & area & " " & volume)
End Sub
```

8. Solution

```
Sub Main(args As String())
  Dim charge, tip, tax, total As Double

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

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

  total = charge + tip + tax

  Console.WriteLine(total)
End Sub
```

9. Solution

```
Sub Main(args As String())
  Dim a, t, s As Double

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

  s = 0.5 * a * t ^ 2

  Console.WriteLine(s)
End Sub
```

10. Solution

```
Sub Main(args As String())
  Dim f, c As Double

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

  c = 5 / 9 * (f - 32)

  Console.WriteLine(c)
End Sub
```

11. Solution

```
Sub Main(args As String())
  Dim w, h As Integer
  Dim bmi As Double

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

```
  bmi = w * 703.0 / h ^ 2

  Console.WriteLine(bmi)
End Sub
```

12. Solution

```
Sub Main(args As String())
  Dim s_total, g_rate, tip, total As Double

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

  tip = s_total * g_rate / 100

  total = s_total + tip

  Console.WriteLine("Tip is $" & tip & " and total is $" & total)
End Sub
```

13. Solution

```
Const VAT = 0.20

Sub Main(args As String())
  Dim btax_price1, btax_price2, btax_price3, atax_price1, atax_price2, atax_price3, avg As Double

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

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

  avg = (atax_price1 + atax_price2 + atax_price3) / 3

  Console.WriteLine(avg)
End Sub
```

14. Solution

```
Const VAT = 0.20

Sub Main(args As String())
  Dim atax_price, btax_price As Double

  Console.Write("Enter after-tax price: ")
  atax_price = Console.ReadLine()
```

```
  btax_price = atax_price / (1 + VAT)

  Console.WriteLine(btax_price)
End Sub
```

15. Solution

```
Sub Main(args As String())
  Dim i_price, discount, f_price, saved As Double

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

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

  Console.WriteLine(f_price & " " & saved)
End Sub
```

16. Solution

```
Const VAT = 0.20

Sub Main(args As String())
  Dim i_kWh, f_kWh, kWh_consumed As Integer
  Dim cost As Double

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

  kWh_consumed = f_kWh - i_kWh

  cost = kWh_consumed * 0.06
  cost += cost * VAT

  Console.WriteLine(kWh_consumed & " " & cost)
End Sub
```

17. Solution

```
Sub Main(args As String())
  Dim day, month, days_passed, days_left As Integer

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

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

```
    Console.WriteLine(days_left)
End Sub
```

Chapter 11

11.3 Review Questions: True/False

1. True
2. False
3. False
4. False
5. False
6. False
7. True
8. True
9. True
10. True
11. False
12. 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 = Console.ReadLine()`	**9.0**	?	?
2	`a += 6 / Math.Sqrt(a) * 2 + 20.4`	**33.4**	?	**?**
3	`b = Math.Round(a) Mod 4`	33.4	**1.0**	**?**
4	`c = b Mod 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 = Console.ReadLine()`	**4.0**	?	?
2	`a += 6 / Math.Sqrt(a) * 2 + 20.4`	**30.4**	?	**?**
3	`b = Math.Round(a) Mod 4`	30.4	**2.0**	**?**
4	`c = b Mod 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 = Console.ReadLine()`	**–2**	?	?
2	`b = Math.Abs(a) Mod 4 + a ^ 4`	–2	**18**	**?**
3	`c = b Mod 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 = Console.ReadLine()	**–3**	?	?
2	b = Math.Abs(a) Mod 4 + a ^ 4	–3	**84**	**?**
3	c = b Mod 5	–3	84	**4**
4	Console.WriteLine(b & ", " & c)	It displays: 84, 4		

3. Solution

```
Sub Main(args As String())
  Dim degrees, radians As Double

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

  degrees = radians * 180 / Math.PI

  Console.WriteLine(degrees)
End Sub
```

4. Solution

```
Sub Main(args As String())
  Dim a, b, hypotenuse As Double

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

  hypotenuse = Math.Sqrt(a ^ 2 + b ^ 2)

  Console.WriteLine(hypotenuse)
End Sub
```

5. Solution

```
Sub Main(args As String())
  Dim adjacent, opposite, th As Double

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

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

  Console.WriteLine(opposite)
End Sub
```

Chapter 12

12.2 Review Exercises

1. Solution

i. a, e, g, h

ii. c, f

2. Solution

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

ii. `y = (3 * x ^ 2 - x ^ 3 / 4) ^ (1 / 5)`

iii. `y = Math.Sqrt(x ^ 4 - 2 * x ^ 3 - 7 * x ^ 2 + x) / (4 * (7 * x ^ 4 - 3 / 4 * x ^ 3) * (7 * x ^ 2 + x)) ^ (1 / 3)`

iv. `y = x / (x - 3 * (x - 1)) + x * (x - 1) ^ (1 / 5) / ((x ^ 3 - 2) * (x - 1) ^ 3)`

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

vi. `y = (Math.Sin(Math.PI / 2 * x) + Math.Cos(3 * Math.PI / 2 * w)) ^ 3 / (Math.Tan(2 * Math.PI / 3 * w) - Math.Sin(Math.PI / 2 * x)) ^ 0.5 + 6`

3. Solution

```
Sub Main(args As String())
  Dim x, y As Double

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

  y = Math.Sqrt(x) * (x ^ 3 + x ^ 2)

  Console.WriteLine(y)
End Sub
```

4. Solution

```
Sub Main(args As String())
  Dim x, y As Double

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

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

  Console.WriteLine(y)
End Sub
```

5. Solution

```
Sub Main(args As String())
  Dim w, x, y As Double

  Console.Write("Enter value for x: ")
```

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

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

  Console.WriteLine(y)
End Sub
```

6. Solution

```
Sub Main(args As String())
  Dim w, x, y As Double

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

  y = (3 + w) / (6 * x - 7 * (x + 4)) + x * (3 * w + 1) ^ (1 / 5) * (5 * x + 4) / ((x ^ 3 + 3) * (x - 1) ^ 7)

  Console.WriteLine(y)
End Sub
```

7. Solution

```
Sub Main(args As String())
  Dim w, x, y As Double

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

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

  Console.WriteLine(y)
End Sub
```

8. Solution

```
Sub Main(args As String())
  Dim a, b, c, area, semi As Double

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

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

```
  Console.WriteLine(area)
End Sub
```

Chapter 13

13.2 Review Exercises

1. Solution

```
Sub Main(args As String())
  Dim last_digit, n, result As Integer

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

  last_digit = n Mod 10
  result = last_digit * 8

  Console.WriteLine(result)
End Sub
```

2. Solution

```
Sub Main(args As String())
  Dim digit1, digit2, digit3, digit4, digit5, number, r, reversed_number As Integer

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

  digit5 = number Mod 10
  r = number \ 10

  digit4 = r Mod 10
  r = number \ 10

  digit3 = r Mod 10
  r = number \ 10

  digit2 = r Mod 10
  digit1 = number \ 10

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

3. Solution

```
Sub Main(args As String())
  Dim n, result As Integer

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

  result = n Mod 2

  Console.WriteLine(result)
End Sub
```

4. Solution

```
Sub Main(args As String())
  Dim n, result As Integer

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

  result = 1 - n Mod 2

  Console.WriteLine(result)
End Sub
```

5. Solution

```
Sub Main(args As String())
  Dim days, hours, minutes, number, r, seconds, weeks As Integer

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

  weeks = number \ 604800  ' 60 * 60 * 24 * 7 = 604800
  r = number Mod 604800

  days = r \ 86400         ' 60 * 60 * 24 = 86400
  r = r Mod 86400

  hours = r \ 3600
  r = r Mod 3600

  minutes = r \ 60
  seconds = r Mod 60

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

6. Solution

```
Sub Main(args As String())
  Dim amount, r, usd1, usd10, usd20, usd5 As Integer

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

  usd20 = amount \ 20
  r = amount Mod 20

  usd10 = r \ 10
  r = r Mod 10

  usd5 = r \ 5
  usd1 = r Mod 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")
End Sub
```

7. Solution

```
Sub Main(args As String())
  Dim distance, feet, inches, miles, r, steps, yards As Integer

  Console.Write("Enter number of steps: ")
  steps = Console.ReadLine()

  distance = steps * 25

  miles = distance \ 63360
  r = distance Mod 63360

  yards = r \ 36
  r = r Mod 36

  feet = r \ 12
  inches = r Mod 12

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

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

14.5 Review Questions: Multiple Choice

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

14.6 Review Exercises

1. Solution

```
Sub Main(args As String())
  Dim first_name, last_name, middle_name, title As String

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

  Console.WriteLine(title & " " & first_name & " " & middle_name & " " & last_name)
  Console.WriteLine(first_name & " " & middle_name & " " & last_name)
  Console.WriteLine(last_name & ", " & first_name)
  Console.WriteLine(last_name & ", " & first_name & " " & middle_name)
  Console.WriteLine(last_name & ", " & first_name & " " & middle_name & ", " & title)
  Console.WriteLine(first_name & " " & last_name)
End Sub
```

2. Solution

```
Sub Main(args As String())
  Dim alphabet, random_word As String

  Dim rnd As New Random()

  alphabet = "abcdefghijklmnopqrstuvwxyz"

  random_word = (alphabet(rnd.Next(0, 26))).ToUpper() &
                alphabet(rnd.Next(0, 26)) &
                alphabet(rnd.Next(0, 26)) &
                alphabet(rnd.Next(0, 26)) &
                alphabet(rnd.Next(0, 26))

  Console.WriteLine(random_word)
End Sub
```

3. Solution

```
Sub Main(args As String())
  Dim name, x, secret_password As String
  Dim rnd As New Random()

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

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

```
  secret_password = x(rnd.Next(0, x.Length)) &
                    x(rnd.Next(0, x.Length)) &
                    x(rnd.Next(0, x.Length)) &
                    rnd.Next(1000, 10000)

  Console.WriteLine(secret_password)
End Sub
```

4. Solution

First approach

```
Sub Main(args As String())
  Dim number, reversed_number As Integer
  Dim s_number, digit1, digit2, digit3 As String

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

  s_number = number.ToString()

  digit1 = s_number(0)
  digit2 = s_number(1)
  digit3 = s_number(2)

  reversed_number = 100 * Int32.Parse(digit3) + 10 * Int32.Parse(digit2) + Int32.Parse(digit1)

  Console.WriteLine(reversed_number)
End Sub
```

Second approach

```
Sub Main(args As String())
  Dim number, reversed_number As Integer
  Dim s_number As String

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

  s_number = number.ToString()
  reversed_number = Int32.Parse(s_number(2) & s_number(1) & s_number(0))

  Console.WriteLine(reversed_number)
End Sub
```

Review in “Sequence Control Structures”

Review Crossword Puzzle

1.

Chapter 15

15.9 Review Questions: True/False

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

15.10 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

Boolean Expression1 (BE1)	Boolean Expression2 (BE2)	BE1 Or BE2	BE1 And BE2	Not(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 Or c > b And c > 1	a > 3 And c > b Or c > 1
4	**–6**	**2**	True	True

−3	**2**	**−4**	False	False
2	**5**	**5**	False	True

5. Solution

Expression	Value
`(x + y) ^ 3`	8
`(x + y) / (x ^ 2 - 14)`	1
`x - 1 = y + 5`	True
`x > 2 And y = 1`	False
`x = 1 Or y = -2 And Not(flag = False)`	True
`Not(x >= 3) And (x Mod 2 > 1)`	False

6. Solution

i. False
ii. True

7. Solution

i. `age < 12 And age <> 8`
ii. `age >= 6 And age <= 9 Or age = 11`
iii. `age > 7 And age <> 10 And age <> 12`
iv. `age = 6 Or age = 9 Or age = 11`
v. `age >= 6 And age <= 12 And age <> 8`
vi. `age <> 7 And age <> 10`

8. Solution

i. `x <> 4 Or y = 3`
ii. `x + 4 > 0`
iii. `Not(x <= 5) And y <> 4`
iv. `x = False`
v. `Not(x < 4 And z <= 4)`
vi. `x = 2 Or x < -5`

9. Solution

i. `Not(x < 4 Or y = 10)`
ii. `Not(x - 2 < 9)`
iii. `Not(Not(x < 2) And y = 4)`
iv. `Not(x = False And y <> 3)`
v. First approach: `Not(Not(x < 2 Or y < 2))`
Second approach: `x < 2 Or y < 2`
vi. `Not(x = -2 Or 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. d

16.4 Review Exercises

1. Solution

The corrections/additions are in red

```
Sub Main(args As String())
  Dim x, y As Double

  x = Console.ReadLine()

  y = -5
  If x * y / 2 > 20 Then
    y *= 2
    x += 4 * x ^ 2
  End If

  Console.WriteLine(x, y)
End Sub
```

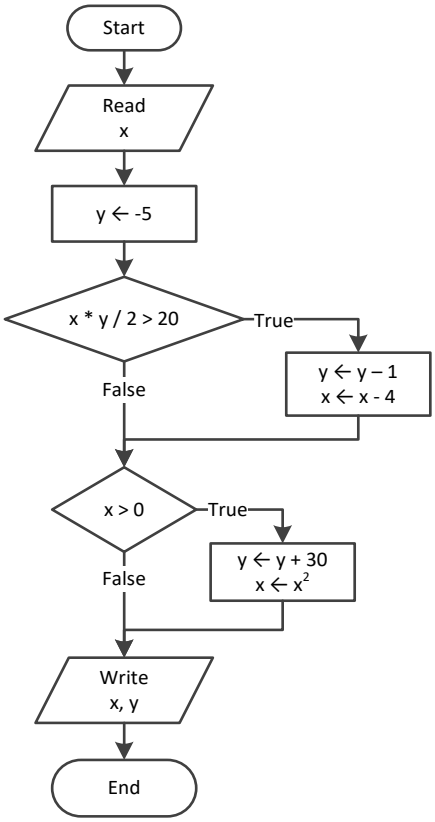
2. Solution

For the input value of 10

Step	Statement	x	y
1	`x = Console.ReadLine()`	**10.0**	?
2	`y = -5`	10.0	**–5.0**
3	`If x * y / 2 > 20 Then`	False	
4	`If x > 0 Then`	True	
5	`y += 30`	10.0	**25.0**
6	`x = 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 = Console.ReadLine()`	**–10.0**	?
2	`y = -5`	–10.0	**–5.0**
3	`If x * y / 2 > 20 Then`	True	
4	`y -= 1`	–10.0	**–6.0**
5	`x -= 4`	**–14.0**	–6.0
6	`If x > 0 Then`	False	
7	`Console.WriteLine(x & ", " & y)`	It displays: –14, –6	



3. *Solution*

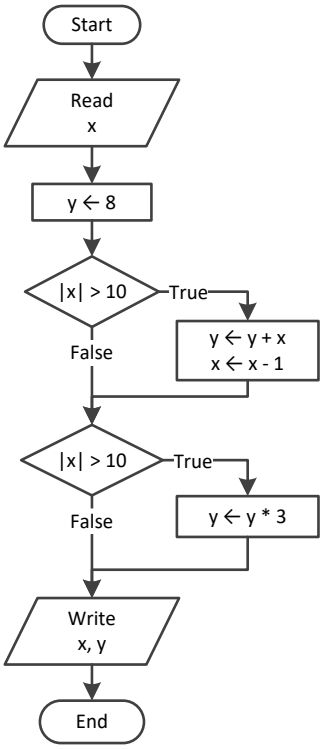
For the input value of –11

Step	Statement	x	y
1	`x = Console.ReadLine()`	**–11**	?
2	`y = 8`	–11	**8**
3	`If Math.Abs(x) > 10 Then`	True	
4	`y += x`	–11	**–3**
5	`x -= 1`	**–12**	–3
6	`If Math.Abs(x) > 10 Then`	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 = Console.ReadLine()`	**11**	?
2	`y = 8`	11	**8**
3	`If Math.Abs(x) > 10 Then`	True	
4	`y += x`	11	**19**
5	`x -= 1`	**10**	19
6	`If Math.Abs(x) > 10 Then`	False	
7	`Console.WriteLine(x & ", " & y)`	It displays: 10, 19	



4. *Solution*

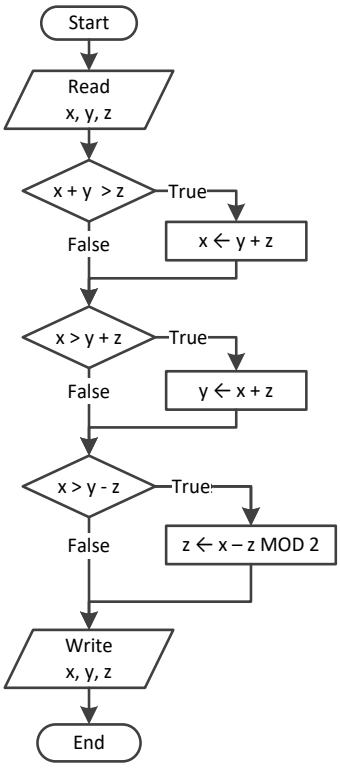
For input values of 1, 2 and 3

Step	Statement	x	y	z
1	`x = Console.ReadLine()`	**1**	?	?
2	`y = Console.ReadLine()`	1	**2**	?
3	`z = Console.ReadLine()`	1	**2**	**3**
4	`If x + y > z Then`	False		
5	`If x > y + z Then`	False		
6	`If x > y - z Then`	True		
7	`z = x - z Mod 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 = Console.ReadLine()`	**4**	?	?
2	`y = Console.ReadLine()`	4	**2**	?
3	`z = Console.ReadLine()`	4	**2**	**1**
4	`If x + y > z Then`		True	
5	`x = y + z`	**3**	2	1
6	`If x > y + z Then`		False	
7	`If x > y - z Then`		True	
8	`z = x - z Mod 2`	3	2	**2**
9	`Console.WriteLine(x & ", " & y & ", " & z)`	It displays: 3, 2, 2		



5. Solution

```
Sub Main(args As String())
  Dim x As Double

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

  If x > 0 Then
    Console.WriteLine("Positive")
  End If
End Sub
```

6. Solution

```
Sub Main(args As String())
  Dim x, y As Double

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

  If x > 0 And y > 0 Then
    Console.WriteLine("Positive")
  End If
End Sub
```

7. Solution

```
Sub Main(args As String())
  Dim x, y As Double

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

  If x > 14 Then
    Console.WriteLine("You can drive a car in Kansas (USA)")
  End If
End Sub
```

8. Solution

```
Sub Main(args As String())
  Dim str As String

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

  If str = str.ToUpper() Then
    Console.WriteLine("Uppercase")
  End If
End Sub
```

9. Solution

```
Sub Main(args As String())
  Dim str As String

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

  If str.Length > 20 Then
    Console.WriteLine("Many characters")
  End If
End Sub
```

10. Solution

```
Sub Main(args As String())
  Dim n1, n2, n3, n4 As Double

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

  If n1 < 0 Or n2 < 0 Or n3 < 0 Or n4 < 0 Then
    Console.WriteLine("Among the given numbers, there is a negative one!")
  End If
End Sub
```

11. Solution

```
Sub Main(args As String())
  Dim a, b, c As Double

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

  If a > b Then
    c = a
    a = b
    b = c
  End If

  Console.WriteLine(a & ", " & b)
End Sub
```

12. Solution

```
Sub Main(args As String())
  Dim average, t1, t2, t3 As Double

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

  average = (t1 + t2 + t3) / 3

  If average > 60 Then
```

```
    Console.WriteLine("Heat Wave")
  End If
End Sub
```

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

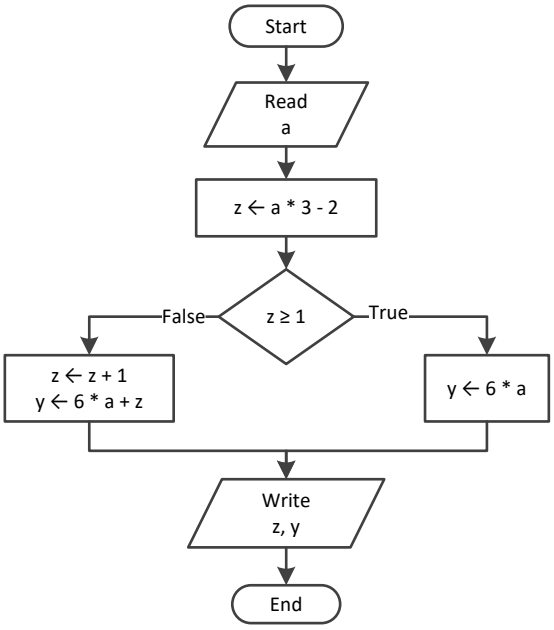
1. Solution

For input value of 3

Step	Statement	a	y	z
1	`a = Console.ReadLine()`	**3.0**	?	?
2	`z = a * 3 - 2`	3.0	?	**7.0**
3	`If z >= 1 Then`		True	
4	`y = 6 * a`	3.0	**18.0**	7.0
5	`Console.WriteLine(z & ", " & y)`	It displays: 7 18		

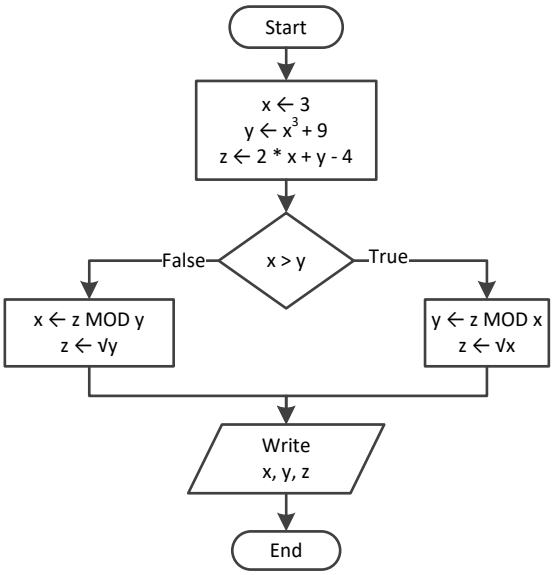
For input value of 0.5

Step	Statement	a	y	z
1	`a = Console.ReadLine()`	**0.5**	?	?
2	`z = a * 3 - 2`	0.5	?	**–0.5**
3	`If z >= 1 Then`		False	
4	`z += 1`	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 = x ^ 3 + 9`	3.0	**36.0**	?
3	`z = 2 * x + y - 4`	3.0	36.0	**38.0**
4	`If x > y Then`		False	
5	`x = z Mod 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

```
Sub Main(args As String())
  Dim w, x, y, z As Double

  x = Console.ReadLine()
  w = x * 3 - 15
  z = (w + 7) * (x + 4) - 10
  If w > x And z > x Then
    x += 1
    y = x / 2 + 4
  Else
    y = x / 4 + 2
  End If
  Console.WriteLine(y)
End Sub
```

For input value of 10

Step	Statement	x	y	w	z
1	x = 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 And z > x Then	True			
5	x += 1	**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 = 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 And z > x Then	False			
5	y = x / 4 + 2	2.0	**2.5**	–9.0	–22.0
6	Console.WriteLine(y)	It displays: 2.5			

4. Solution

```
Sub Main(args As String())
  Dim name1, name2 As String
  Dim goals1, goals2 As Integer

  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 = Console.ReadLine()
  Console.Write("Enter goals " & name2 & " scored: ")
  goals2 = Console.ReadLine()

  If goals1 > goals2 Then
    Console.WriteLine("Winner: " & name1)
  Else
    Console.WriteLine("Winner: " & name2)
  End If
End Sub
```

5. Solution

```
Sub Main(args As String())
  Dim x As Integer

  x = Console.ReadLine()
  If x Mod 6 = 0 Then
    Console.WriteLine(x & " is a multiple of 6")
  Else
    Console.WriteLine(x & " is not a multiple of 6")
  End If
End Sub
```

6. Solution

```
Sub Main(args As String())
  Dim x As Integer

  x = Console.ReadLine()
  If x Mod 6 = 0 Or x Mod 7 = 0 Then
    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")
  End If
End Sub
```

7. Solution

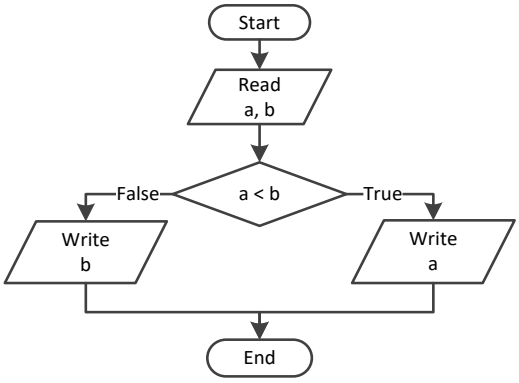
```
Sub Main(args As String())
  Dim x, y As Integer

  x = Console.ReadLine()

  y = x Mod 4
  If y = 0 Then
    Console.WriteLine(x & " is a multiple of 4")
  Else
    Console.WriteLine(x & " is not a multiple of 4")
  End If

  Console.WriteLine("The structure is: " & x & " = " & (x \ 4) & " x 4 + " & y)
```

```
End Sub
```

8. Solution

```
Sub Main(args As String())
  Dim x As Integer

  x = Console.ReadLine()

  If x >= 1000 And x <= 9999 Then
    Console.WriteLine(x & " is a four-digit integer")
  Else
    Console.WriteLine(x & " is not a four-digit integer")
  End If
End Sub
```

9. Solution




```
Sub Main(args As String())
  Dim a, b As Double

  a = Console.ReadLine()
  b = Console.ReadLine()

  If a < b Then
    Console.WriteLine(a)
  Else
    Console.WriteLine(b)
  End If
End Sub
```

10. Solution

```
Sub Main(args As String())
  Dim a, b, c As Double

  a = Console.ReadLine()
  b = Console.ReadLine()
  c = Console.ReadLine()

  If a < b + c And b < a + c And c < a + b Then
    Console.WriteLine("Given numbers can be lengths of the three sides of a triangle")
```

```
  Else
    Console.WriteLine("Given numbers cannot be lengths of the three sides of a triangle")
  End If
End Sub
```

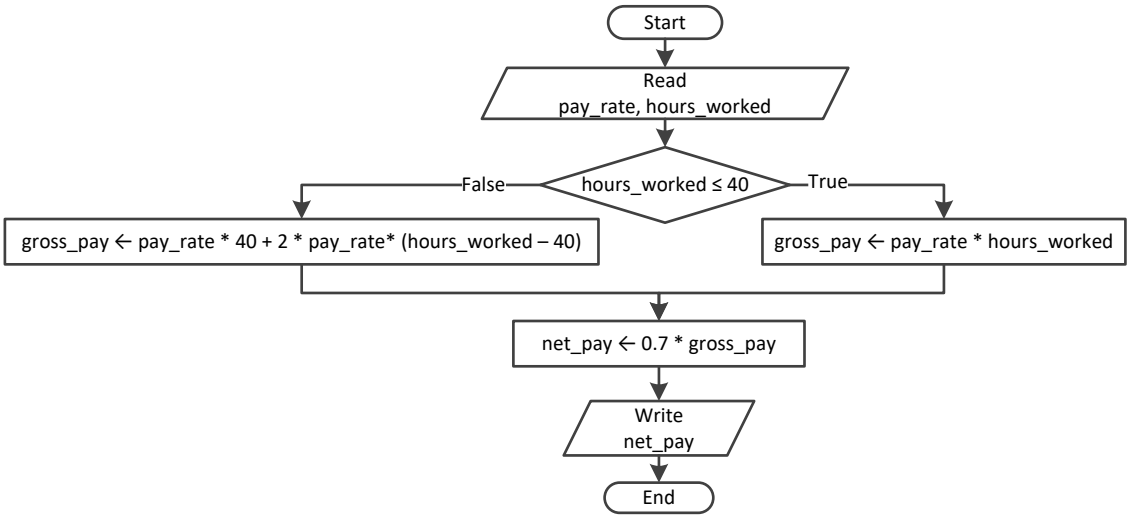
11. Solution

```
Sub Main(args As String())
  Dim a, b, c As Double

  a = Console.ReadLine()
  b = Console.ReadLine()
  c = Console.ReadLine()

  If a ^ 2 = b ^ 2 + c ^ 2 Or b ^ 2 = a ^ 2 + c ^ 2 Or c ^ 2 = a ^ 2 + b ^ 2 Then
    Console.WriteLine("Given numbers can be lengths of the three sides of a right triangle")
  Else
    Console.WriteLine("Given numbers cannot be lengths of the three sides of a right triangle")
  End If
End Sub
```

12. Solution

```
Sub Main(args As String())
  Dim a, average, b, c As Double

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

  average = (a + b + c) / 3

  If average >= 8 Then
    Console.WriteLine("Qualified")
  Else
    Console.WriteLine("Disqualified")
  End If
End Sub
```

13. Solution




```
Sub Main(args As String())
  Dim gross_pay, net_pay, pay_rate As Double
  Dim hours_worked As Integer

  pay_rate = Console.ReadLine()
  hours_worked = Console.ReadLine()

  If hours_worked <= 40 Then
    gross_pay = pay_rate * hours_worked
  Else
    gross_pay = pay_rate * 40 + 2 * pay_rate * (hours_worked - 40)
  End If

  net_pay = 0.7 * gross_pay
  Console.WriteLine(net_pay)
End Sub
```

14. Solution

```
Sub Main(args As String())
  Dim miles, miles_left, r As Integer

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

  r = miles Mod 12000

  If r > 6000 Then
    miles_left = 12000 - r
    Console.WriteLine("Your car needs a major service in " & miles_left & " miles")
  Else
    miles_left = 6000 - r
    Console.WriteLine("Your car needs a minor service in " & miles_left & " miles")
  End If
End Sub
```

15. Solution

```
Sub Main(args As String())
  Dim a1, a2, s1, s2, t As Double

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

  s1 = 0.5 * a1 * t ^ 2
  s2 = 0.5 * a2 * t ^ 2

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

  If s1 > s2 Then
    Console.WriteLine("Car A is first")
  Else
    Console.WriteLine("Car B is first")
  End If
End Sub
```

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 = Console.ReadLine()`	**5**	?
2	`If q > 0 And q <= 50 Then`	True	
3	`b = 1`	5	**1**
4	`Console.WriteLine(b)`	It displays: 1	

For input value of 150

Step	Statement	q	b
1	`q = Console.ReadLine()`	**150**	?
2	`If q > 0 And q <= 50 Then`	False	
3	`ElseIf q > 50 And q <= 100 Then`	False	
4	`ElseIf q > 100 And q <= 200 Then`	True	
5	`b = 3`	150	**3**
6	`Console.WriteLine(b)`	It displays: 3	

For input value of 250

Step	Statement	q	b
1	`q = Console.ReadLine()`	**250**	?
2	`If q > 0 And q <= 50 Then`	False	
3	`ElseIf q > 50 And q <= 100 Then`	False	
4	`ElseIf q > 100 And q <= 200 Then`	False	
5	`b = 4`	250	**4**
6	`Console.WriteLine(b)`	It displays: 4	

For input value of −1

Step	Statement	q	b
1	`q = Console.ReadLine()`	**−1**	?
2	`If q > 0 And q <= 50 Then`	False	
3	`ElseIf q > 50 And q <= 100 Then`	False	
4	`ElseIf q > 100 And q <= 200 Then`	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 = Console.ReadLine()	**5.0**	?	?
2	discount = 0	5.0	**0.0**	?
3	If amount < 20 Then		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 = Console.ReadLine()	**150.0**	?	?
2	discount = 0	150.0	**0.0**	?
3	If amount < 20 Then		False	
4	ElseIf amount >=20 And amount < 60 Then		False	
5	ElseIf amount >= 60 And amount < 100 Then		False	
6	ElseIf amount >= 100 Then		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 = Console.ReadLine()	**−1.0**	?	?
2	discount = 0	−1.0	**0.0**	?
3	If amount < 20 Then		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

```
Sub Main(args As String())
  Dim a, y As Double

  a = Console.ReadLine()

  If a < 1 Then
```

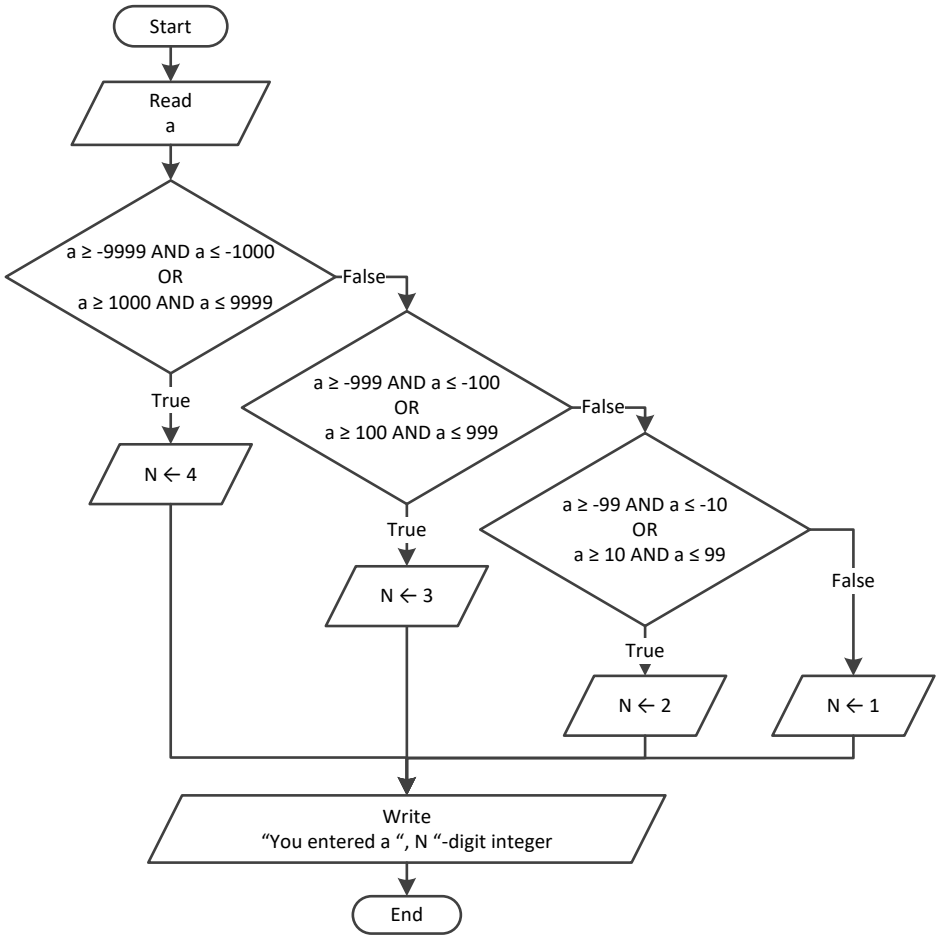
```
    y = 5 + a
    Console.WriteLine(y)
  ElseIf a < 5 Then
    y = 23 / a
    Console.WriteLine(y)
  ElseIf a < 10 Then
    y = 5 * a
    Console.WriteLine(y)
  Else
    Console.WriteLine("Error!")
  End If
End Sub
```

4. Solution

```
Sub Main(args As String())
  Dim name1, name2 As String
  Dim goals1, goals2 As Integer

  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 = Console.ReadLine()
  Console.Write("Enter goals " & name2 & " scored: ")
  goals2 = Console.ReadLine()

  If goals1 > goals2 Then
    Console.WriteLine("Winner: " & name1)
  ElseIf goals2 > goals1 Then
    Console.WriteLine("Winner: " & name2)
  Else
    Console.WriteLine("It's a tie!")
  End If
End Sub
```

5. Solution

First approach




```
Sub Main(args As String())
  Dim a, n As Integer

  a = Console.ReadLine()

  If a >= -9999 And a <= -1000 Or a >= 1000 And a <= 9999 Then
    n = 4
  ElseIf a >= -999 And a <= -100 Or a >= 100 And a <= 999 Then
    n = 3
  ElseIf a >= -99 And a <= -10 Or a >= 10 And a <= 99 Then
    n = 2
  Else
    n = 1
  End If

  Console.WriteLine("You entered a " & n & "-digit integer")
End Sub
```

Second approach

```
Sub Main(args As String())
  Dim a As Integer
  Dim a_string As String
```

```
  a = Console.ReadLine()

  a_string = Math.Abs(a).ToString()
  Console.WriteLine("You entered a " & a_string.Length & "-digit integer")
End Sub
```

6. Solution

First approach

```
Sub Main(args As String())
  Dim a, n As Integer

  a = Console.ReadLine()

  If a >= -9999 And a <= -1000 Or a >= 1000 And a <= 9999 Then
    Console.WriteLine("You entered a 4-digit integer")
  ElseIf a >= -999 And a <= -100 Or a >= 100 And a <= 999 Then
    Console.WriteLine("You entered a 3-digit integer")
  ElseIf a >= -99 And a <= -10 Or a >= 10 And a <= 99 Then
    Console.WriteLine("You entered a 2-digit integer")
  ElseIf a >= -9 And a <= 9 Then   'Include the value of zero
    Console.WriteLine("You entered a 1-digit integer")
  Else
    Console.WriteLine("Error: Invalid value!")
  End If
End Sub
```

Second approach

```
Sub Main(args As String())
  Dim a As Integer
  Dim a_string As String

  a = Console.ReadLine()

  If a >= -9999 And a <= 9999 Then
    a_string = Math.Abs(a).ToString()
    Console.WriteLine("You entered a " & a_string.Length & "-digit integer")
  Else
    Console.WriteLine("Error: Invalid value!")
  End If
End Sub
```

7. Solution

```
Sub Main(args As String())
  Dim cad, eur, gbp, jpy, usd As Double
  Dim ch As Integer

  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 = Console.ReadLine()

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

  If ch = 1 Then
    eur = usd * 0.87
    Console.WriteLine("$" & usd & " = " & eur & " EUR")
  ElseIf ch = 2 Then
    gbp = usd * 0.78
    Console.WriteLine("$" & usd & " = " & gbp & " GBP")
  ElseIf ch = 3 Then
    jpy = usd * 108.55
    Console.WriteLine("$" & usd & " = " & jpy & " JPY")
  Else
    cad = usd * 1.33
    Console.WriteLine("$" & usd & " = " & cad & " CAD")
  End If
End Sub
```

8. Solution

```
Sub Main(args As String())
  Dim m As Integer

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

  If m <= 2 Or m = 12 Then
    Console.WriteLine("Winter")
  ElseIf m <= 5 Then
    Console.WriteLine("Spring")
  ElseIf m <= 8 Then
    Console.WriteLine("Summer")
  Else
    Console.WriteLine("Fall (Autumn)")
  End If
End Sub
```

9. Solution

```
Sub Main(args As String())
  Dim m As Integer

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

  If m < 1 Or m > 12 Then
    Console.WriteLine("Error: Invalid value!")
  ElseIf m <= 2 Or m = 12 Then
    Console.WriteLine("Winter")
  ElseIf m <= 5 Then
```

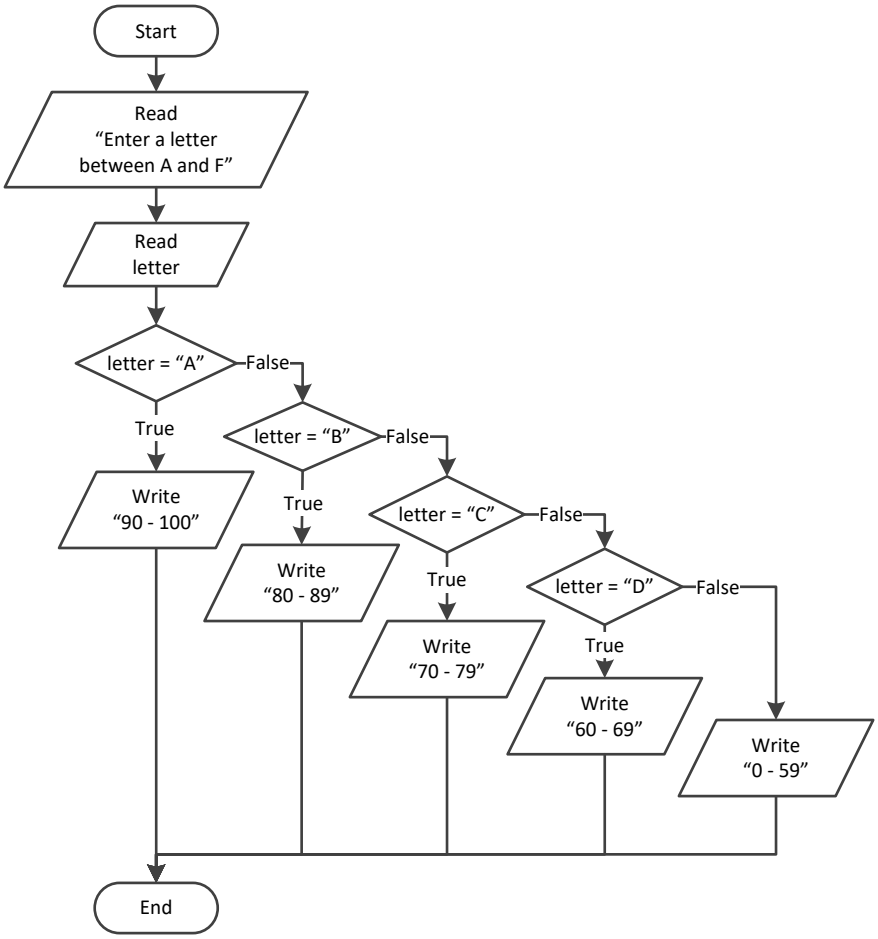
```
    Console.WriteLine("Spring")
  ElseIf m <= 8 Then
    Console.WriteLine("Summer")
  Else
    Console.WriteLine("Fall (Autumn)")
  End If
End Sub
```

10. Solution

```
Sub Main(args As String())
  Dim n As Double
  Dim x, y As Integer

  Console.Write("Enter a number between 1.0 and 4.9: ")
  n = Console.ReadLine()

  x = Fix(n)
  y = Fix(n * 10) Mod 10

  If x = 1 Then
    Console.Write("One")
  ElseIf x = 2 Then
    Console.Write("Two")
  ElseIf x = 3 Then
    Console.Write("Three")
  ElseIf x = 4 Then
    Console.Write("Four")
  End If

  Console.Write(" point ")

  If y = 1 Then
    Console.WriteLine("one")
  ElseIf y = 2 Then
    Console.WriteLine("two")
  ElseIf y = 3 Then
    Console.WriteLine("three")
  ElseIf y = 4 Then
    Console.WriteLine("four")
  ElseIf y = 5 Then
    Console.WriteLine("five")
  ElseIf y = 6 Then
    Console.WriteLine("six")
  ElseIf y = 7 Then
    Console.WriteLine("seven")
  ElseIf y = 8 Then
    Console.WriteLine("eight")
  ElseIf y = 9 Then
    Console.WriteLine("nine")
  ElseIf y = 0 Then
    Console.WriteLine("zero")
```

```
  End If
End Sub
```

11. Solution




```
Sub Main(args As String())
  Dim letter As String

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

  If letter = "A" Then
    Console.WriteLine("90 - 100")
  ElseIf letter = "B" Then
    Console.WriteLine("80 - 89")
  ElseIf letter = "C" Then
    Console.WriteLine("70 - 79")
  ElseIf letter = "D" Then
    Console.WriteLine("60 - 69")
  Else
    Console.WriteLine("0 - 59")
  End If
End Sub
```

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 = 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 = 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 = 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 += 1`	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 = Console.ReadLine()`	**10**	?	?
2	`x = Console.ReadLine()`	10	**2**	?
3	`y = Console.ReadLine()`	10	2	**5.0**
4	`Case a = 10`	True		
5	`x = x Mod 2`	10	**0**	5.0
6	`y = 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 = Console.ReadLine()`	**5**	?	?
2	`x = Console.ReadLine()`	5	**2**	?
3	`y = 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 = Console.ReadLine()`	**4**	?	?
2	`x = Console.ReadLine()`	4	**6**	?
3	`y = 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 += 1`	4	3	**3.0**
9	`Console.WriteLine(x & ", " & y)`	It displays: 3, 3		

3. Solution

```
Sub Main(args As String())
  Dim name As String

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

  Select Case name
    Case "January"
      Console.WriteLine("1")
    Case "February"
      Console.WriteLine("2")
    Case "March"
      Console.WriteLine("3")
    Case "April"
      Console.WriteLine("4")
    Case "May"
      Console.WriteLine("5")
    Case "June"
      Console.WriteLine("6")
    Case "July"
      Console.WriteLine("7")
    Case "August"
      Console.WriteLine("8")
    Case "September"
      Console.WriteLine("9")
    Case "October"
      Console.WriteLine("10")
    Case "November"
      Console.WriteLine("11")
    Case "December"
      Console.WriteLine("12")
    Case Else
      Console.WriteLine("Error")
  End Select
End Sub
```

4. Solution

```
Sub Main(args As String())
  Dim choice As Integer
  Dim feet, inches, miles, yards As Double

  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 = Console.ReadLine()

  Select Case choice
```

```
    Case 1
      Console.Write("Enter miles: ")
      miles = Console.ReadLine()
      yards = miles * 1760
      Console.WriteLine(miles & " miles = " & yards & " yards")
    Case 2
      Console.Write("Enter miles: ")
      miles = Console.ReadLine()
      feet = miles * 5280
      Console.WriteLine(miles & " miles = " & feet & " feet")
    Case 3
      Console.Write("Enter miles: ")
      miles = Console.ReadLine()
      inches = miles * 63360
      Console.WriteLine(miles & " miles = " & inches & " inches")
    Case Else
      Console.WriteLine("Invalid choice!")
  End Select
End Sub
```

5. Solution

```
Sub Main(args As String())
  Dim roman As String

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

  Select Case roman
    Case "I"
      Console.WriteLine("1")
    Case "II"
      Console.WriteLine("2")
    Case "III"
      Console.WriteLine("3")
    Case "IV"
      Console.WriteLine("4")
    Case "V"
      Console.WriteLine("5")
    Case "VI"
      Console.WriteLine("6")
    Case "VII"
      Console.WriteLine("7")
    Case "VIII"
      Console.WriteLine("8")
    Case "IX"
      Console.WriteLine("9")
    Case "X"
      Console.WriteLine("10")
    Case Else
      Console.WriteLine("Error")
  End Select
```

```
End Sub
```

6. Solution

```
Sub Main(args As String())
  Dim total As Integer

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

  Select Case total
    Case 1
      Console.WriteLine("You are awarded 3 points")
    Case 2
      Console.WriteLine("You are awarded 10 points")
    Case 3
      Console.WriteLine("You are awarded 20 points")
    Case Else
      Console.WriteLine("You are awarded 45 points")
  End Select
End Sub
```

7. Solution

```
Sub Main(args As String())
  Dim i As Integer
  Dim name As String

  Dim rnd As New Random()

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

  i = rnd.Next(0, 3)

  Select Case i
    Case 0
      Console.WriteLine("Good morning " & name)
    Case 1
      Console.WriteLine("Good evening " & name)
    Case 2
      Console.WriteLine("Good night " & name)
  End Select
End Sub
```

8. Solution

```
Sub Main(args As String())
  Dim num As String

  num = Console.ReadLine()

  Select Case num
    Case "zero"
```

```
      Console.WriteLine(0)
    Case "one"
      Console.WriteLine(1)
    Case "two"
      Console.WriteLine(2)
    Case "three"
      Console.WriteLine(3)
    Case "four"
      Console.WriteLine(4)
    Case "five"
      Console.WriteLine(5)
    Case "six"
      Console.WriteLine(6)
    Case "seven"
      Console.WriteLine(7)
    Case "eight"
      Console.WriteLine(8)
    Case "nine"
      Console.WriteLine(9)
    Case Else
      Console.WriteLine("I don't know this number!")
  End Select
End Sub
```

9. Solution

```
Sub Main(args As String())
  Dim b As Integer

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

  Select Case b
    Case 0
      Console.WriteLine("Calm")
    Case 1
      Console.WriteLine("Light Air")
    Case 2
      Console.WriteLine("Light breeze")
    Case 3
      Console.WriteLine("Gentle breeze")
    Case 4
      Console.WriteLine("Moderate breeze")
    Case 5
      Console.WriteLine("Fresh breeze")
    Case 6
      Console.WriteLine("Strong breeze")
    Case 7
      Console.WriteLine("Moderate gale")
    Case 8
      Console.WriteLine("Gale")
    Case 9
```

```
      Console.WriteLine("Strong gale")
    Case 10
      Console.WriteLine("Storm")
    Case 11
      Console.WriteLine("Violent storm")
    Case 12
      Console.WriteLine("Hurricane force")
    Case Else
      Console.WriteLine("Invalid Beaufort number!")
  End Select
End Sub
```

Chapter 20

20.3 Review Questions: True/False

1. True
2. True
3. False
4. False
5. True

20.4 Review Exercises

1. Solution

For input values of 20, 1

Step	Statement	x	y
1	`x = Console.ReadLine()`	**20**	?
2	`y = Console.ReadLine()`	20	**1**
3	`If x < 30 Then`	True	
4	`Case y = 1`	True	
5	`x = x Mod 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 = Console.ReadLine()`	**20**	?
2	`y = Console.ReadLine()`	20	**3**
3	`If x < 30 Then`	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 = Console.ReadLine()`	**12**	?
2	`y = Console.ReadLine()`	12	**8**
3	`If x < 30 Then`	True	
4	`Case y = 1`	False	
5	`Case y = 2`	False	
6	`Case y = 3`	False	
7	`x -= 2`	**10**	8

8	`y += 1`	10	**9**
9	`Console.WriteLine(x & ", " & y)`	It displays: 10, 9	

For input values of 50, 0

Step	Statement	x	y
1	`x = Console.ReadLine()`	**50**	?
2	`y = Console.ReadLine()`	50	**0**
3	`If x < 30 Then`	False	
4	`y += 1`	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 = Console.ReadLine()`	**60**	?
2	`y = Console.ReadLine()`	60	25
3	`If (x + y) / 2 <= 20 Then`	False	
4	`If y < 15 Then`	False	
5	`ElseIf y < 23 Then`	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 = Console.ReadLine()`	**50**	?
2	`y = Console.ReadLine()`	50	8
3	`If (x + y) / 2 <= 20 Then`	False	
4	`If y < 15 Then`	True	
5	`x = x Mod 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 = Console.ReadLine()`	**20**	?
2	`y = Console.ReadLine()`	20	15
3	`If (x + y) / 2 <= 20 Then`	True	
4	`If y < 10 Then`	False	
5	`ElseIf y < 20 Then`	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 = Console.ReadLine()`	**10**	?
2	`y = Console.ReadLine()`	10	30
3	`If (x + y) / 2 <= 20 Then`	True	
4	`If y < 10 Then`	False	
5	`ElseIf y < 20 Then`	False	
6	`x = x - 2`	**8**	30
7	`y += 3`	8	**33**
8	`Console.WriteLine(x & ", " & y)`	It displays: 8, 33	

3. Solution

```
Sub Main(args As String())
  Dim a As Integer

  a = Console.ReadLine()

  If a > 1000 Then
    Console.WriteLine("Big Positive")
  Else
    If a > 0 Then
      Console.WriteLine("Positive")
    Else
      If a < -1000 Then
        Console.WriteLine("Big Negative")
      Else
        If a < 0 Then
          Console.WriteLine("Negative")
        Else
          Console.WriteLine("Zero")
        End If
      End If
    End If
  End If
End Sub
```

4. Solution

```
Sub Main(args As String())
  Dim a, b, c As Double

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

```
  b = Console.ReadLine()
  c = Console.ReadLine()

  If a >= b + c Or b >= a + c Or c >= a + b Then
    Console.WriteLine("Given numbers cannot be lengths of the three sides of a triangle")
  Else
    If a = b And b = c Then
      Console.WriteLine("Equilateral")
    ElseIf a ^ 2 = b ^ 2 + c ^ 2 Or b ^ 2 = a ^ 2 + c ^ 2 Or c ^ 2 = a ^ 2 + b ^ 2 Then
      Console.WriteLine("Right (or right-angled)")
    Else
      Console.WriteLine("Not special")
    End If
  End If
End Sub
```

5. Solution

```
Sub Main(args As String())
  Dim amount, pin, r, usd1, usd10, usd5 As Integer

  Console.Write("Enter your four-digit PIN : ")
  pin = Console.ReadLine()
  If pin <> 1234 Then
    Console.Write("Wrong PIN. Enter your four-digit PIN : ")
    pin = Console.ReadLine()
    If pin <> 1234 Then
      Console.Write("Wrong PIN. Enter your four-digit PIN : ")
      pin = Console.ReadLine()
    End If
  End If

  If pin <> 1234 Then
    Console.WriteLine("PIN locked!")
  Else
    Console.Write("Enter the amount of money (an integer value) that you want to withdraw: ")
    amount = Console.ReadLine()
    usd10 = amount \ 10
    r = amount Mod 10
    usd5 = r \ 5
    usd1 = r Mod 5
    Console.Write(usd10 & " note(s) of $10 " & usd5 & " note(s) of $5 ")
    Console.WriteLine("and " & usd1 & " note(s) of $1")
  End If
End Sub
```

6. Solution

First approach

```
Sub Main(args As String())
  Dim t, w As Double
```

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

  If t > 75 Then
    If w > 12 Then
      Console.WriteLine("The day is hot and windy")
    Else
      Console.WriteLine("The day is hot and not windy")
    End If
  Else
    If w > 12 Then
      Console.WriteLine("The day is cold and windy")
    Else
      Console.WriteLine("The day is cold and not windy")
    End If
  End If
End Sub
```

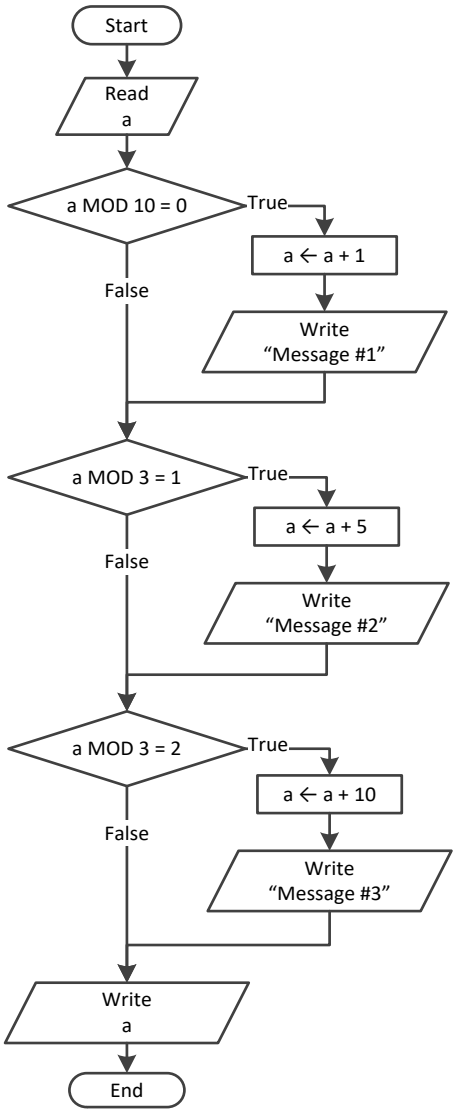
Second approach

```
Sub Main(args As String())
  Dim t, w As Double
  Dim message1, message2 As String

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

  If t > 75 Then
    message1 = "hot"
  Else
    message1 = "cold"
  End If

  If w > 12 Then
    message2 = "windy"
  Else
    message2 = "not windy"
  End If

  Console.WriteLine("The day is " & message1 & " and " & message2)
End Sub
```

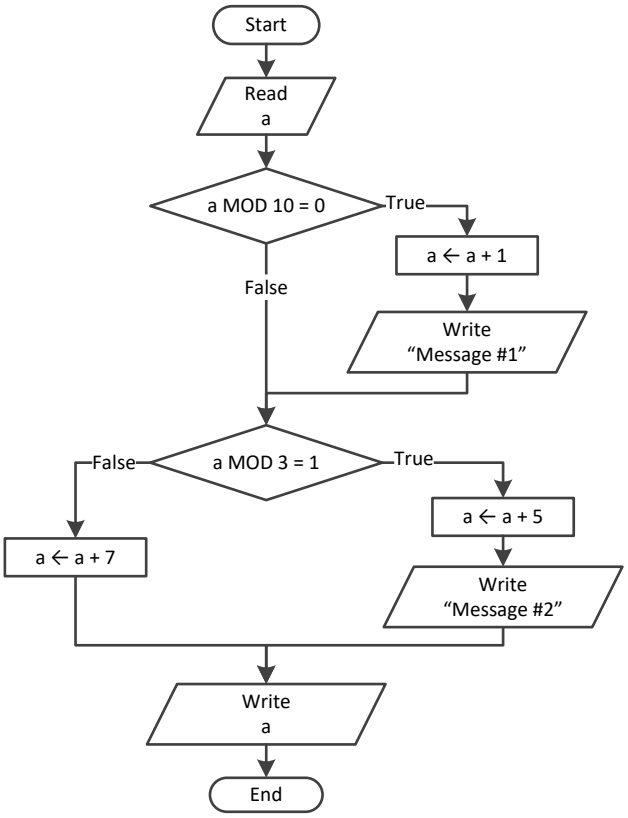
Chapter 21

21.4 Review Exercises

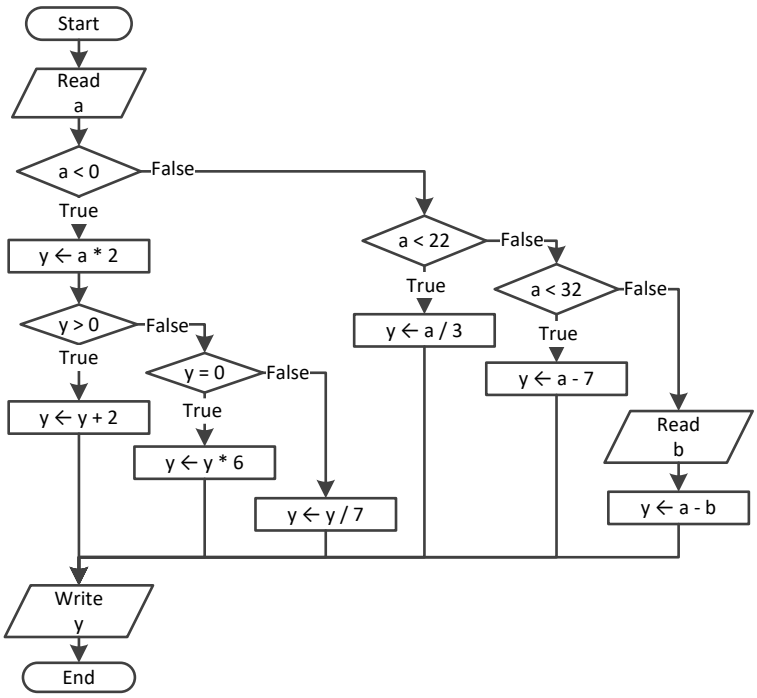
1. Solution



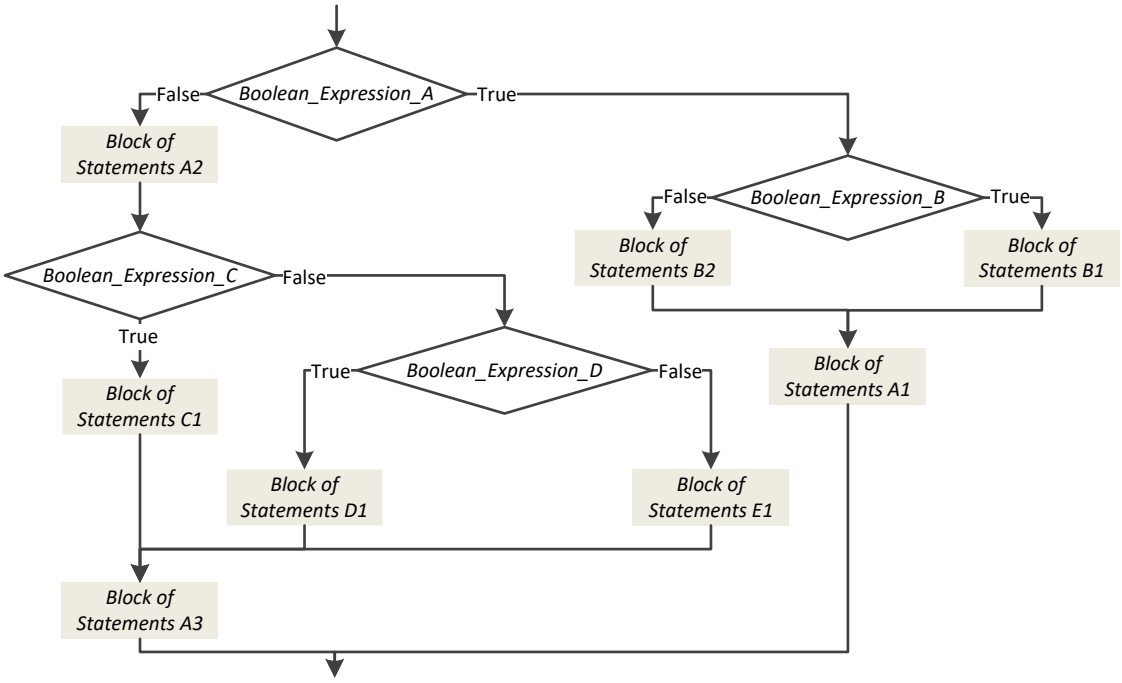
2. Solution



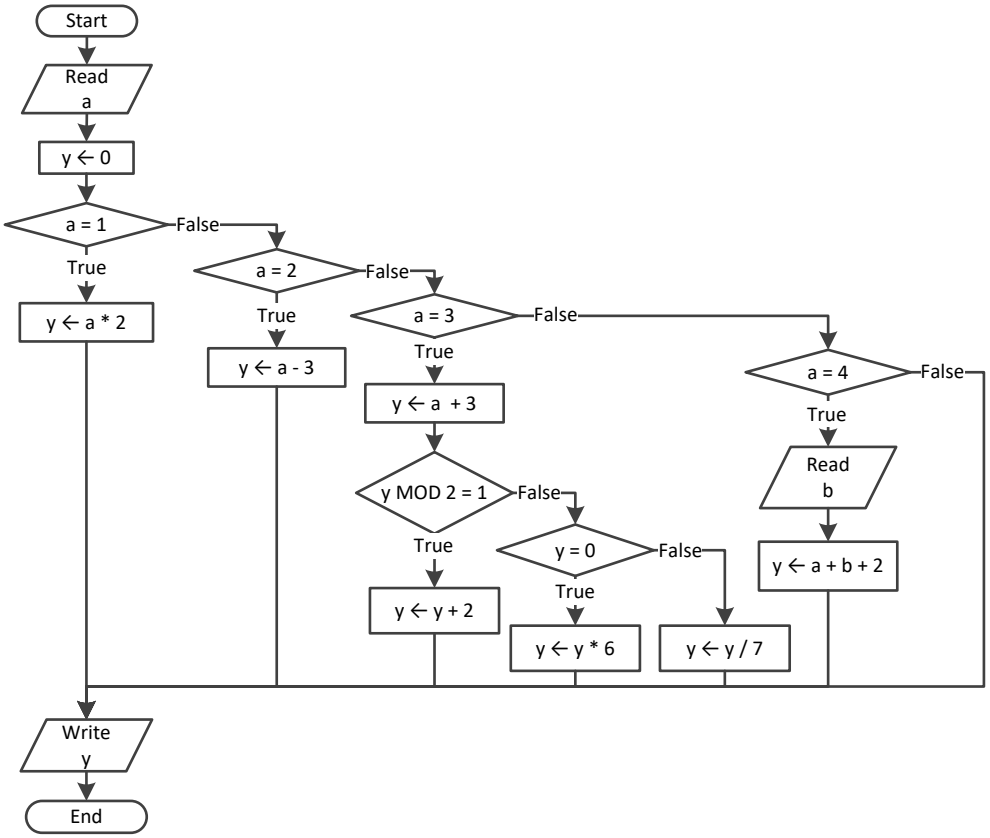
3. Solution



4. Solution



5. Solution



6. Solution

```
Sub Main(args As String())
  Dim x, y, z As Double

  x = Console.ReadLine()
  y = Console.ReadLine()

  If x <> 100 Or y <= 10 Then
    z = Console.ReadLine()
    If z <= x + y Then
      x -= 3
      y = x + 4
    End If
  End If
  Console.WriteLine(x & " " & y)
End Sub
```

7. Solution

```
Sub Main(args As String())
  Dim x As Integer

  x = Console.ReadLine()

  If x = 1 Then
    Console.WriteLine("Good Morning")
    Console.WriteLine("How do you do?")
    Console.WriteLine("Is everything okay?")
  ElseIf x = 2 Then
    Console.WriteLine("Good Evening")
    Console.WriteLine("How do you do?")
    Console.WriteLine("Is everything okay?")
  ElseIf x = 3 Then
    Console.WriteLine("Good Afternoon")
    Console.WriteLine("Is everything okay?")
  Else
    Console.WriteLine("Good Night")
  End If
End Sub
```

8. Solution

```
Sub Main(args As String())
  Dim a, b, c, d, y As Integer

  a = Console.ReadLine()
  b = Console.ReadLine()

  c = a Mod 2
  d = b \ 5

  If a >= b Then
```

```
    y = 1
  ElseIf d > c And a > 2 Then
    y = 2
  ElseIf d * c > a / b Then
    If d * c > 10 Then
      y = 4
    Else
      y = 3
    End If
  Else
    y = 5
  End If

  Console.WriteLine(y)
End Sub
```

9. Solution

```
Sub Main(args As String())
  Dim x As Integer

  x = Console.ReadLine()

  If x > 0 Then
    If x Mod 10 = 0 Then
      Console.WriteLine("Last digit equal to 0")
    ElseIf x Mod 10 = 1 Then
      Console.WriteLine("Last digit equal to 1")
    Else
      Console.WriteLine("None")
    End If
  Else
    If x = -1 Then
      Console.WriteLine("Bye")
    Else
      Console.WriteLine("Invalid Number")
    End If
  End If
End Sub
```

10. Solution

```
Sub Main(args As String())
  Dim a, b, y As Double

  a = Console.ReadLine()
  b = Console.ReadLine()

  y = a * b

  If y > 0 Then
    y -= 1
    y /= 2
```

```
  Else
    y +=10
    If y > 0 Then
      y /= 2
    Else
      y *= 2
    End If
  End If
End Sub
```

11. Solution

```
Sub Main(args As String())
  Dim a, b, c As Double

  a = Console.ReadLine()
  b = Console.ReadLine()
  c = Console.ReadLine()

  c = a * b + c
  If c > 0 Then
    c /= 2
    If a > b Then
      a *= 2
      b *= 2
    Else
      c /= 20
      If c <= 10 Then
        b *= 2
      End If
    End If
  Else
    c /= 3
    c /= 20
    If c <= 10 Then
      b *= 2
    End If
  End If
  Console.WriteLine(a & " " & b  & " " & c)
End Sub
```

Chapter 22

22.9 Review Questions: True/False

1. False
2. True
3. False
4. True
5. True
6. False
7. False

22.10 Review Questions: Multiple Choice

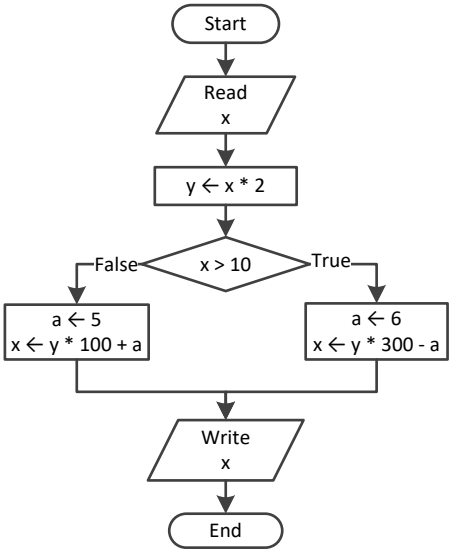
1. a
2. b
3. a
4. c

22.11 Review Exercises

1. Solution

```
Sub Main(args As String())
  Dim a, x, y As Integer

  y = Console.ReadLine()
  x = Console.ReadLine()

  If y > 0 Then
    a = x * 4 * y + 1
  Else
    a = x * 2 * y + 6
  End If
  Console.WriteLine(y)
  Console.WriteLine(a)
End Sub
```

2. Solution



3. Solution

```
Sub Main(args As String())
  Dim a, y As Double

  a = Console.ReadLine()

  If a >= 10 Then
    Console.WriteLine("Error!")
  Else
    If a < 1 Then
      y = 5 + a
    ElseIf a < 5 Then
      y = 23 / a
```

```
    Else
      y = 5 * a
    End If
    Console.WriteLine(y)
  End If
End Sub
```

4. Solution

```
Sub Main(args As String())
  Dim day, month As Integer
  Dim name As String

  day = Console.ReadLine()
  month = Console.ReadLine()
  name = Console.ReadLine()

  If day = 16 And month = 2 And name = "Loukia" Then
    Console.WriteLine("Happy Birthday!!!")
  Else
    Console.WriteLine("No match!")
  End If
End Sub
```

5. Solution

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

```
Sub Main(args As String())
  Dim a, b, c, d As Double

  a = Console.ReadLine()
  b = Console.ReadLine()
  c = Console.ReadLine()

  If a > 10 Then
    If c < 2000 Then
      d = (a + b + c) / 12
      Console.WriteLine("The result is: " & d)
    Else
      Console.WriteLine("Error!")
    End If
  Else
    Console.WriteLine("Error!")
  End If
End Sub
```

6. Solution

```
Sub Main(args As String())
  Dim a, b, c, d As Double

  a = Console.ReadLine()
  b = Console.ReadLine()
```

```
  c = Console.ReadLine()

  If a > 10 And b < 2000 And c <> 10 Then
    d = (a + b + c) / 12
    Console.WriteLine("The result is: " & d)
  End If

  If a <= 10 Then
    Console.WriteLine("Error!")
  End If
End Sub
```

7. Solution

```
Sub Main(args As String())
  Dim a, b, y As Integer

  a = Console.ReadLine()
  b = Console.ReadLine()

  y = 3
  If a > 0 Then
    y = y * a
    Console.WriteLine("Hello Zeus")
  End If

  Console.WriteLine(y & " " & b)
End Sub
```

8. Solution

```
Sub Main(args As String())
  Dim a, b, y As Double

  a = Console.ReadLine()
  b = Console.ReadLine()

  y = 0
  If a > 0 Then
    y = y + 7
  Else
    Console.WriteLine("Hello Zeus")
    Console.WriteLine(Math.Abs(a))
  End If
  Console.WriteLine(y)
End Sub
```

9. Solution

```
Sub Main(args As String())
  Dim os As String

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

```
  If os = "iOS" Then
    Console.WriteLine("Apple")
  ElseIf os = "Android" Then
    Console.WriteLine("Google")
  ElseIf os = "Windows" Then
    Console.WriteLine("Microsoft")
  End If
End Sub
```

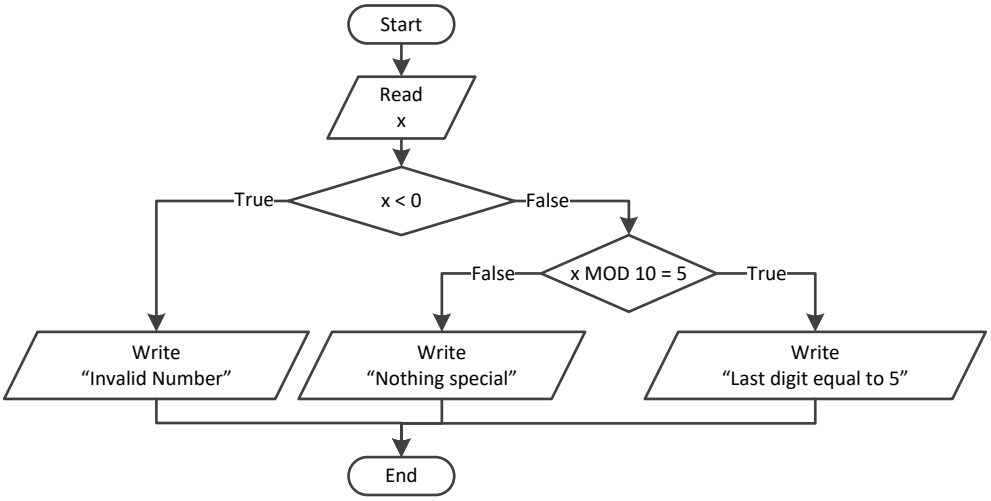
Chapter 23

23.6 Review Exercises

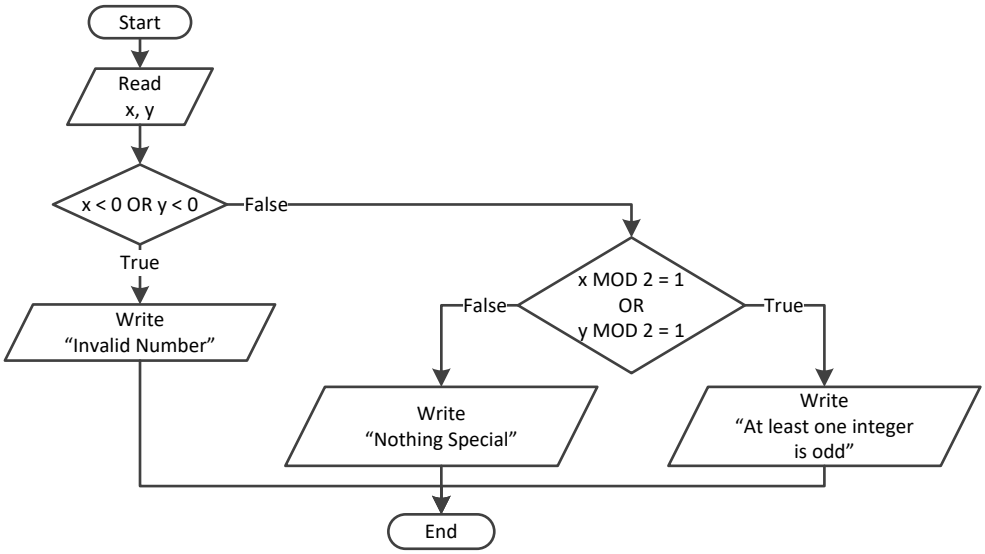
1. Solution

```
Sub Main(args As String())
  Dim x As Double

  Console.Write("Enter a non-negative number: ")
  x = Console.ReadLine()
  If x < 0 Then
    Console.WriteLine("Error! You entered a negative value")
  Else
    Console.WriteLine("The square root of " & x & " is " & Math.Sqrt(x))
  End If
End Sub
```

2. Solution

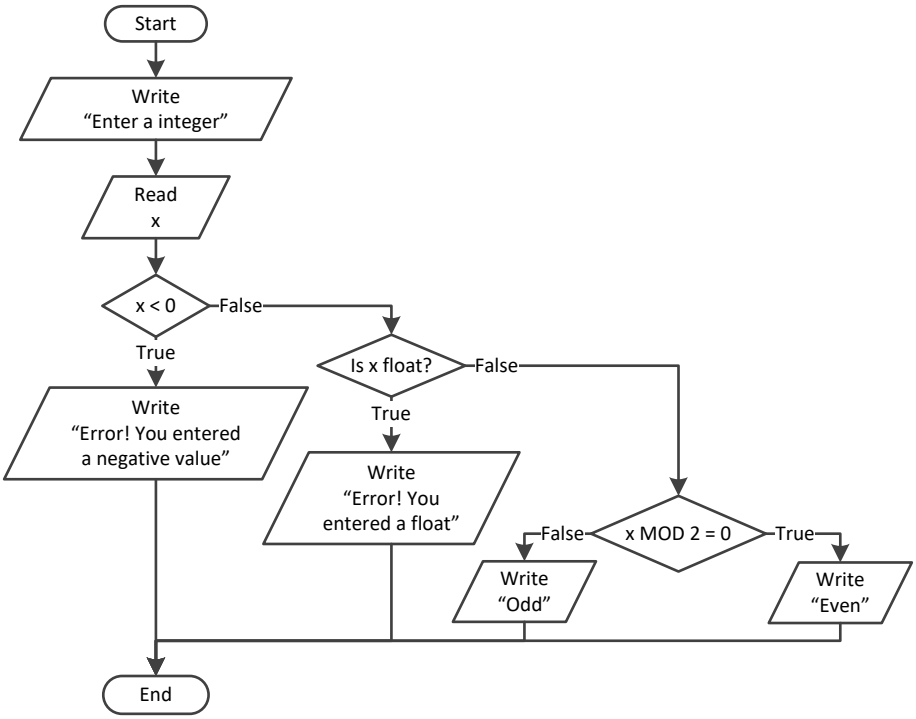


3. Solution



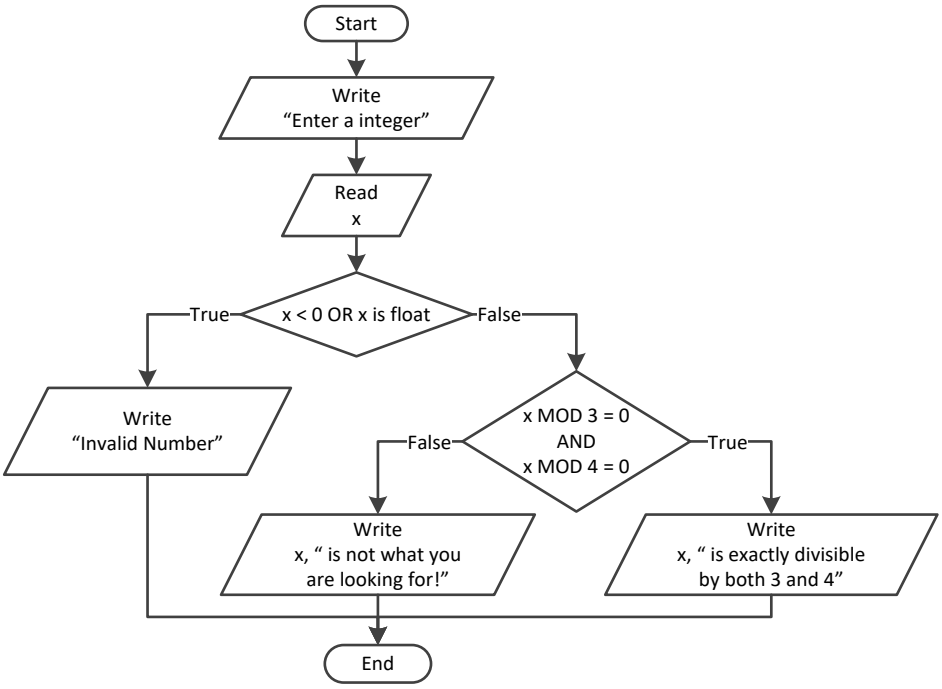

```
Sub Main(args As String())
  Dim x, y As Integer

  x = Console.ReadLine()
  y = Console.ReadLine()

  If x < 0 Or y < 0 Then
    Console.WriteLine("Invalid Number")
  Else
    If x Mod 2 = 1 Or y Mod 2 = 1 Then
      Console.WriteLine("At least one integer is odd")
    Else
      Console.WriteLine("Nothing Special")
    End If
  End If
End Sub
```

4. Solution



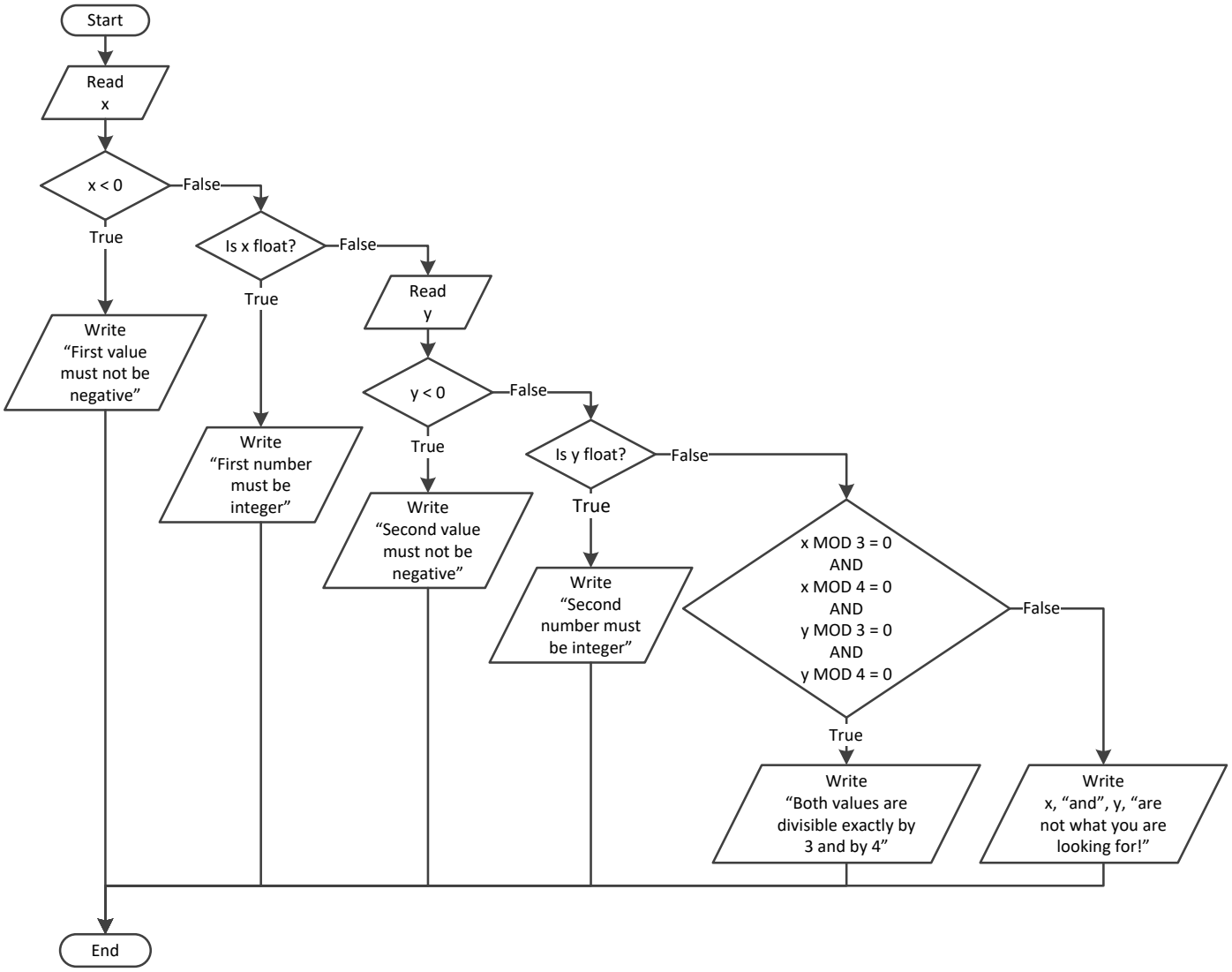

```
Sub Main(args As String())
  Dim x As Double

  Console.Write("Enter a non-negative number: ")
  x = Console.ReadLine()
  If x < 0 Then
    Console.WriteLine("Error! You entered a negative value")
  ElseIf x <> Fix(x) Then
    Console.WriteLine("Error! You entered a float")
  ElseIf x Mod 2 = 0 Then
    Console.WriteLine("Even")
  Else
    Console.WriteLine("Odd")
  End If
End Sub
```

5. Solution




```
Sub Main(args As String())
  Dim x As Double

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

  If x < 0 Or x <> Fix(x) Then
    Console.WriteLine("Invalid Number")
  ElseIf x Mod 3 = 0 And x Mod 4 = 0 Then
    Console.WriteLine(x & " is exactly divisible by both 3 and 4")
  Else
    Console.WriteLine(x & " is not what you are looking for!")
  End If
End Sub
```

6. Solution




```
Sub Main(args As String())
  Dim x, y As Double

  x = Console.ReadLine()

  If x < 0 Then
    Console.WriteLine("First value must be not be negative")
  Else
    If x <> Fix(x) Then
        Console.WriteLine("First number must be integer")
    Else
      y = Console.ReadLine()
      If y < 0 Then
        Console.WriteLine("Second value must be not be negative")
      Else
        If y <> Fix(y) Then
          Console.WriteLine("Second number must be integer")
```

```
        Else
          If x Mod 3 = 0 And x Mod 4 = 0 And y Mod 3 = 0 And y Mod 4 = 0  Then
            Console.WriteLine("Both values are divisible exactly by 3 and by 4")
          Else
            Console.WriteLine("Nothing Special")
          End If
        End If
      End If
    End If
  End If
End Sub
```

7. Solution

```
Sub Main(args As String())
  Dim choice As Integer
  Dim t As Double

  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 = Console.ReadLine()
  Console.Write("Enter a temperature: ")
  t = Console.ReadLine()

  If choice < 1 Or choice > 4 Then
    Console.WriteLine("Wrong choice")
  Else
    Select Case choice
      Case 1
        If t < 0 Then  'Absolute zero in Kelvin
          Console.WriteLine("Wrong temperature")
        Else
          Console.WriteLine(1.8 * t - 459.67)
        End If
      Case 2
        If t < -459.67 Then  'Absolute zero in Fahrenheit
          Console.WriteLine("Wrong temperature")
        Else
          Console.WriteLine((t + 459.57) / 1.8)
        End If
      Case 3
        If t < -459.67 Then  'Absolute zero in Fahrenheit
          Console.WriteLine("Wrong temperature")
        Else
          Console.WriteLine(5 / 9 * (t - 32))
        End If
      Case 4
        If t < -273.15 Then  'Absolute zero in Celcius
```

```
          Console.WriteLine("Wrong temperature")
        Else
          Console.WriteLine(9 / 5 * t + 32)
        End If
    End Select
  End If
End Sub
```

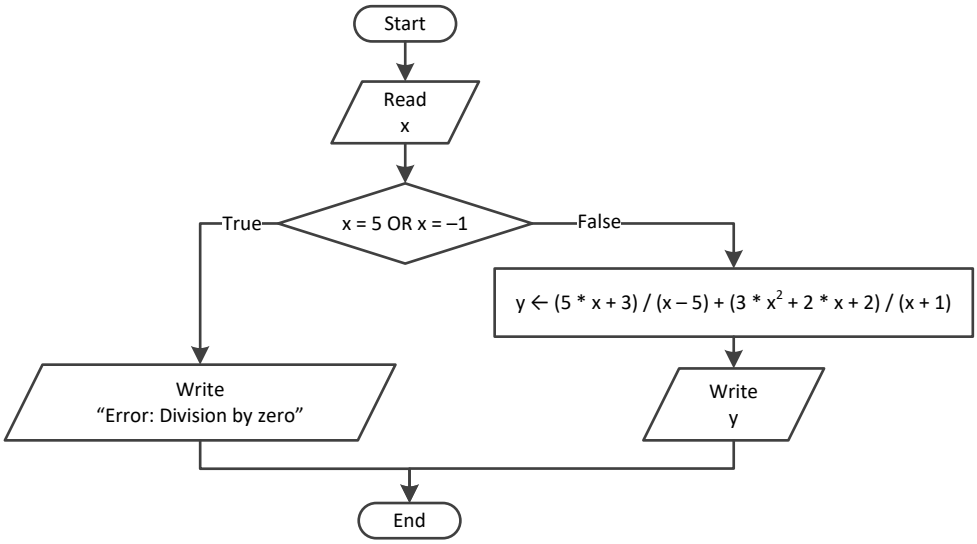
8. Solution

```
Sub Main(args As String())
  Dim a, b As Integer
  Dim op As String

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

  Select Case op
    Case "+"
      Console.WriteLine(a + b)
    Case "-"
      Console.WriteLine(a - b)
    Case "*"
      Console.WriteLine(a * b)
    Case "/"
      If b = 0 Then
        Console.WriteLine("Error: Division by zero")
      Else
        Console.WriteLine(a / b)
      End If
    Case "DIV"
      If b = 0 Then
        Console.WriteLine("Error: Division by zero")
      Else
        Console.WriteLine(a \ b)
      End If
    Case "MOD"
      If b = 0 Then
        Console.WriteLine("Error: Division by zero")
      Else
        Console.WriteLine(a Mod b)
      End If
    Case "POWER"
      Console.WriteLine(a ^ b)
  End Select
End Sub
```

9. Solution

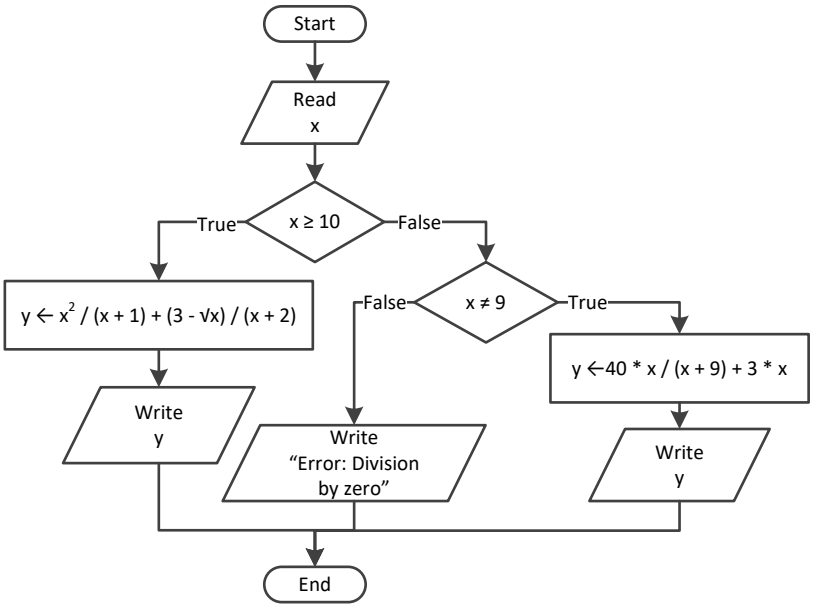
```
Sub Main(args As String())
  Dim a, b As Integer
  Dim op As String

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

  Select Case op
    Case "+"
      Console.WriteLine(a + b)
    Case "-"
      Console.WriteLine(a - b)
    Case "*"
      Console.WriteLine(a * b)
    Case "/"
      If b = 0 Then
        Console.WriteLine("Error: Division by zero")
      Else
        Console.WriteLine(a / b)
      End If
    Case "DIV"
      If b = 0 Then
        Console.WriteLine("Error: Division by zero")
      Else
        Console.WriteLine(a \ b)
      End If
    Case "MOD"
      If b = 0 Then
        Console.WriteLine("Error: Division by zero")
      Else
        Console.WriteLine(a Mod b)
      End If
    Case "POWER"
      Console.WriteLine(a ^ b)
    Case Else
      Console.WriteLine("Error: Invalid operator")
  End Select
End Sub
```

10. Solution




```
Sub Main(args As String())
  Dim x, y As Double

  x = Console.ReadLine()

  If x = 5 Or x = -1 Then
    Console.WriteLine("Error: Division by zero")
  Else
    y = (5 * x + 3) / (x - 5) + (3 * x ^ 2 + 2 * x + 2) / (x + 1)
    Console.WriteLine(y)
  End If
End Sub
```

11. Solution




```
Sub Main(args As String())
```

```
  Dim x, y As Double

  x = Console.ReadLine()
  If x >= 10 Then
    y = x ^ 2 / (x + 1) + (3 - Math.Sqrt(x)) / (x + 2)
    Console.WriteLine(y)
  ElseIf x <> 9 Then
    y = 40 * x / (x + 9) + 3 * x
    Console.WriteLine(y)
  Else
    Console.WriteLine("Error: Division by zero")
  End If
End Sub
```

12. Solution

```
Sub Main(args As String())
  Dim x, y As Double

  x = Console.ReadLine()
  If x <= -15 Or x > 25 Then
    y = x - 1
    Console.WriteLine(y)
  ElseIf x <= -10 Then
    y = x / Math.Sqrt(x + 30) + (8 + x) ^ 2 / (x + 1)
    Console.WriteLine(y)
  ElseIf x <= 0 Then
    y = Math.Abs(40 * x) / (x - 8)
    Console.WriteLine(y)
  Else
    If x = 9 Then
      Console.WriteLine("Error: Division by zero")
    ElseIf x < 9 Then
      Console.WriteLine("Error: Invalid square root")
    Else
      y = 3 * x / Math.Sqrt(x - 9)
      Console.WriteLine(y)
    End If
  End If
End Sub
```

13. Solution

```
Sub Main(args As String())
  Dim a1, a2, a3, maximum, minimum As Integer
  Dim max_name, min_name, n1, n2, n3 As String

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

```
  a2 = Console.ReadLine()
  Console.Write("Enter the name of the second person: ")
  n2 = Console.ReadLine()
  Console.Write("Enter the age of the third person: ")
  a3 = Console.ReadLine()
  Console.Write("Enter the name of the third person: ")
  n3 = Console.ReadLine()

  minimum = a1
  min_name = n1
  If a2 < minimum Then
    minimum = a2
    min_name = n2
  End If
  If a3 < minimum Then
    minimum = a3
    min_name = n3
  End If

  maximum = a1
  max_name = n1
  If a2 > maximum Then
    maximum = a2
    max_name = n2
  End If
  If a3 > maximum Then
    maximum = a3
    max_name = n3
  End If

  Console.WriteLine(min_name & " " & max_name)
End Sub
```

14. Solution

```
Sub Main(args As String())
  Dim age1, age2, age3, maximum, middle, minimum As Integer

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

  minimum = age1
  If age2 < minimum Then
    minimum = age2
  End If
  If age3 < minimum Then
    minimum = age3
  End If
```

```
  maximum = age1
  If age2 > maximum Then
    maximum = age2
  End If
  If age3 > maximum Then
    maximum = age3
  End If

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

15. Solution

```
Sub Main(args As String())
  Dim a1, a2, a3, maximum, minimum, middle As Integer
  Dim max_name, min_name, n1, n2, n3 As String

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

  minimum = a1
  min_name = n1
  If a2 < minimum Then
    minimum = a2
    min_name = n2
  End If
  If a3 < minimum Then
    minimum = a3
    min_name = n3
  End If

  maximum = a1
  max_name = n1
  If a2 > maximum Then
    maximum = a2
    max_name = n2
  End If
  If a3 > maximum Then
    maximum = a3
    max_name = n3
  End If
```

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

  If Math.Abs(maximum - middle) < Math.Abs(minimum - middle) Then
    Console.WriteLine(max_name)
  Else
    Console.WriteLine(min_name)
  End If
End Sub
```

16. Solution

```
Sub Main(args As String())
  Dim digit1, digit2, digit3, r, total As Integer
  Dim x As Double

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

  If x <> Fix(x) Then
    Console.WriteLine("Error! You must enter an integer")
  ElseIf x < 100 Or x > 999 Then
    Console.WriteLine("Entered integer is not a three-digit integer")
  Else
    digit1 = x \ 100
    r = x Mod 100

    digit2 = r \ 10
    digit3 = r Mod 10

    total = digit1 ^ 3 + digit2 ^ 3 + digit3 ^ 3

    If total = x Then
      Console.WriteLine("You entered an Armstrong number!")
    Else
      Console.WriteLine("You entered a non-Armstrong number!")
    End If
  End If
End Sub
```

17. Solution

```
Sub Main(args As String())
  Dim d, m, y As Integer

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

  If m = 2 Then
    If y Mod 4 = 0 And y Mod 100 <> 0 Or y Mod 400 = 0 Then
```

```
      Console.WriteLine(29 - d)
    Else
      Console.WriteLine(28 - d)
    End If
  ElseIf m = 4 Or m = 6 Or m = 9 Or m = 11 Then
    Console.WriteLine(30 - d)
  Else
    Console.WriteLine(31 - d)
  End If
End Sub
```

18. Solution

First approach

```
Sub Main(args As String())
  Dim word, word1, word2 As String

  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 Or word = word2 Then
    Console.WriteLine("Word is okay!")
  Else
    Console.WriteLine("Word is not okay")
  End If
End Sub
```

Second approach

```
Sub Main(args As String())
  Dim word, word1, word2 As String

  word = Console.ReadLine()

  word1 = word(0).ToUpper() &
          word(1).ToLower() &
          word(2).ToUpper() &
          word(3).ToLower() &
          word(4).ToUpper() &
          word(5).ToLower()
```

```
  word2 = word(0).ToLower() &
          word(1).ToUpper() &
          word(2).ToLower() &
          word(3).ToUpper() &
          word(4).ToLower() &
          word(5).ToUpper()

  If word = word1 Or word = word2 Then
    Console.WriteLine("Word is okay!")
  Else
    Console.WriteLine("Word is not okay")
  End If
End Sub
```

19. Solution

```
Sub Main(args As String())
  Dim q As Integer
  Dim discount, payment As Double

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

  If q < 3 Then
    discount = 0
  ElseIf q < 6 Then
    discount = 10
  ElseIf q < 10 Then
    discount = 15
  ElseIf q < 14 Then
    discount = 20
  ElseIf q < 20 Then
    discount = 27
  Else
    discount = 30
  End If

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

  Console.WriteLine("You got a discount of " & discount & "%")
  Console.WriteLine("You must pay $" & payment)
End Sub
```

20. Solution

```
Const VAT = 0.19

Sub Main(args As String())
  Dim amount, discount, payment As Double

  Console.Write("Enter a before-tax amount: : ")
  amount = Console.ReadLine()
```

```
  If amount < 0 Then
    Console.WriteLine("Error! You entered a negative value")
  Else
    If amount < 50 Then
      discount = 0
    ElseIf amount < 100 Then
      discount = 1
    ElseIf amount < 250 Then
      discount = 2
    Else
      discount = 3
    End If

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

    Console.WriteLine("You got a discount of " & discount & "%")
    Console.WriteLine("You must pay $" & payment)
  End If
End Sub
```

21. Solution

```
Sub Main(args As String())
  Dim a, h, w As Integer
  Dim bmi As Double

  Console.Write("Enter age: ")
  a = Console.ReadLine()
  If a < 18 Then
    Console.WriteLine("Invalid age")
  Else
    Console.Write("Enter weight in pounds: ")
    w = Console.ReadLine()
    Console.Write("Enter height in inches: ")
    h = Console.ReadLine()

    bmi = w * 703 / h ^ 2

    If bmi < 15 Then
      Console.WriteLine("Very severely underweight")
    ElseIf bmi < 16 Then
      Console.WriteLine("Severely underweight")
    ElseIf bmi < 18.5 Then
      Console.WriteLine("Underweight")
    ElseIf bmi < 25 Then
      Console.WriteLine("Normal")
    ElseIf bmi < 30 Then
      Console.WriteLine("Overweight")
    ElseIf bmi < 35 Then
      Console.WriteLine("Severely overweight")
    Else
```

```
      Console.WriteLine("Very severely overweight")
    End If
  End If
End Sub
```

22. Solution

```
Const TAX_RATE = 0.10

Sub Main(args As String())
  Dim water As Integer
  Dim total As Double

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

  If water < 0 Then
    Console.WriteLine("Error! You entered a negative value")
  Else
    If water <= 10 Then
      total = water * 3
    ElseIf water <= 20 Then
      total = 10 * 3 + (water - 10) * 5
    ElseIf water <= 35 Then
      total = 10 * 3 + 10 * 5 + (water - 20) * 7
    Else
      total = 10 * 3 + 10 * 5 + 15 * 7 + (water - 35) * 9
    End If

    total = total + total * TAX_RATE
    Console.WriteLine("Total amount to pay (taxes included): " & total)
  End If
End Sub
```

23. Solution

```
Sub Main(args As String())
  Dim children As Integer
  Dim income, tax As Double

  Console.Write("Enter taxable income: ")
  income = Console.ReadLine()
  Console.Write("Enter number of children: ")
  children = Console.ReadLine()

  If income <= 8000 Then
    tax = income * 0.10
  ElseIf income <= 30000 Then
    tax = 8000 * 0.10 + (income - 8000) * 0.15
  ElseIf income <= 70000 Then
    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
```

```
  End If

  If children > 0 Then
    tax = tax - tax * 0.02
  End If
  Console.WriteLine("Tax: " & tax)
End Sub
```

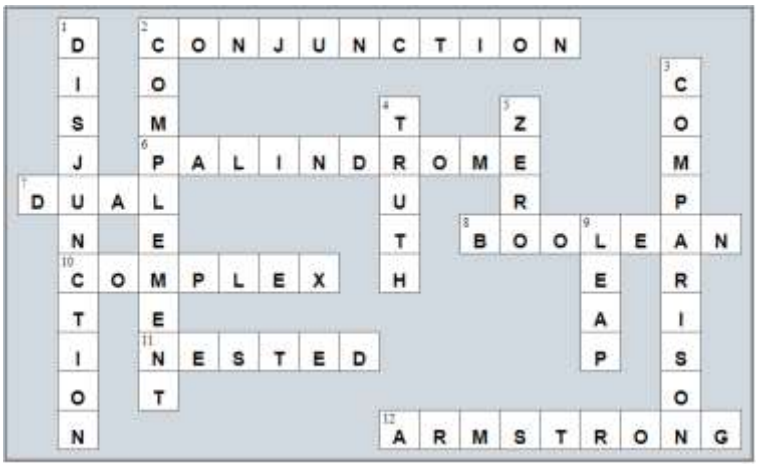
24. Solution

```
Sub Main(args As String())
  Dim wind As Double

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

  If wind < 0 Then
    Console.WriteLine("Error! You entered a negative value")
  Else
    If wind < 1 Then
      Console.WriteLine("Beaufort: 0" & vbCrLf & "Calm")
    ElseIf wind < 4 Then
      Console.WriteLine("Beaufort: 1" & vbCrLf & "Light air")
    ElseIf wind < 8 Then
      Console.WriteLine("Beaufort: 2" & vbCrLf & "Light breeze")
    ElseIf wind < 13 Then
      Console.WriteLine("Beaufort: 3" & vbCrLf & "Gentle breeze")
    ElseIf wind < 18 Then
      Console.WriteLine("Beaufort: 4" & vbCrLf & "Moderate breeze")
    ElseIf wind < 25 Then
      Console.WriteLine("Beaufort: 5" & vbCrLf & "Fresh breeze")
    ElseIf wind < 31 Then
      Console.WriteLine("Beaufort: 6" & vbCrLf & "Strong breeze")
    ElseIf wind < 39 Then
      Console.WriteLine("Beaufort: 7" & vbCrLf & "Moderate gale")
    ElseIf wind < 47 Then
      Console.WriteLine("Beaufort: 8" & vbCrLf & "Gale")
    ElseIf wind < 55 Then
      Console.WriteLine("Beaufort: 9" & vbCrLf & "Strong gale")
    ElseIf wind < 64 Then
      Console.WriteLine("Beaufort: 10" & vbCrLf & "Storm")
    ElseIf wind < 74 Then
      Console.WriteLine("Beaufort: 11" & vbCrLf & "Violent storm")
    Else
      Console.WriteLine("Beaufort: 12" & vbCrLf & "Hurricane force")
    End If

    If wind < 13 Then
      Console.WriteLine("It's Fishing Day!!!")
    End If
  End If
End Sub
```

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. True
9. False
10. False
11. False
12. True
13. False
14. False
15. True
16. 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

```
Sub Main(args As String())
  Dim i As Integer

  i = 3
  Do Loop
    Console.WriteLine(i)
    i -= 1
  Loop While i >= 0
  Console.WriteLine("The end")
End Sub
```

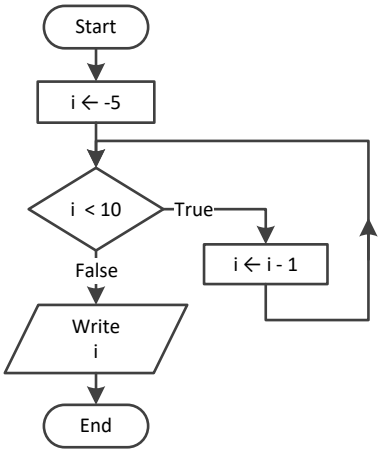
2. Solution

Step	Statement	i	x
1	i = 3	**3**	?
2	x = 0	3	**0**
3	Do While i >= 0	True	
4	i -= 1	**2**	0
5	x += i	2	**2**
6	Do While i >= 0	True	
7	i -= 1	**1**	2
8	x += i	1	**3**

9	`Do While i >= 0`	True	
10	`i -= 1`	**0**	3
11	`x += i`	0	**3**
12	`Do While i >= 0`	True	
13	`i -= 1`	**–1**	3
14	`x += i`	–1	**2**
15	`Do 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	`Do While i < 10`	True	
3	`i -= 1`		**–6**
4	`Do While i < 10`	True	
5	`i -= 1`		**–7**
6	`Do While i < 10`	True	
7	`i -= 1`		**–8**
8	…	**…**	**…**
9	…	**…**	**…**

It performs an infinite number of iterations

4. Solution

Step	Statement	a	b	c	d
1	`a = 2`	**2**	?	?	?
2	`Do 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	`Case d = 4`	True			
7	`Console.WriteLine(b & ", " & c)`	It displays: 3, 6			
8	`a += 4`	**6**	3	6	4
9	`Do 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	`Case d = 4`	False			
14	`Case d = 5`	False			
15	`Case d = 8`	True			
16	`Console.WriteLine(a & ", " & b)`	It displays: 6, 7			
17	`a += 4`	**10**	7	14	8
18	`Do 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	`Case d = 4`	False			
23	`Case d = 5`	False			
24	`Case d = 8`	False			
25	`Console.WriteLine(a & ", " & b & ", " & d)`	It displays: 10, 11, 12			
26	`a += 4`	**14**	11	22	12
27	`Do 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	`Do While b < 2`	True				
6	`x = a + b`	1	1	0	0	**2**
7	`If x Mod 2 <> 0 Then`	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	`Do While b < 2`	True				
13	`x = a + b`	1	0	1	1	**1**
14	`If x Mod 2 <> 0 Then`	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	`Do 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 += 1`	**39**	10
5	`Console.WriteLine(y)`	It displays: 10	
6	`Loop While y < x`	True	
7	`y *= 2`	39	**20**
8	`x += 1`	**40**	20
9	`Console.WriteLine(y)`	It displays: 20	
10	`Loop While y < x`	True	
11	`y *= 2`	40	**40**
12	`x += 1`	**41**	40
13	`Console.WriteLine(y)`	It displays: 40	
14	`Loop While y < x`	True	
15	`y *= 2`	41	**80**
16	`x += 1`	**42**	80
17	`Console.WriteLine(y)`	It displays: 80	

18	`Loop While y < x`	False

8. Solution

Step	Statement	Notes	x
1	`x = 1`		**1**
2	`If x Mod 2 = 0 Then`	False	
3	`x += 3`		**4**
4	`Console.WriteLine(x)`	It displays: 4	
5	`Loop While x < 12`	True	
6	`If x Mod 2 = 0 Then`	True	
7	`x += 1`		**5**
8	`Console.WriteLine(x)`	It displays: 5	
9	`Loop While x < 12`	True	
10	`If x Mod 2 = 0 Then`	False	
11	`x += 3`		**8**
12	`Console.WriteLine(x)`	It displays: 8	
13	`Loop While x < 12`	True	
14	`If x Mod 2 = 0 Then`	True	
15	`x += 1`		**9**
16	`Console.WriteLine(x)`	It displays: 9	
17	`Loop While x < 12`	True	
18	`If x Mod 2 = 0 Then`	False	
19	`x += 3`		**12**
20	`Console.WriteLine(x)`	It displays: 12	
21	`Loop While x < 12`	False	

9. Solution

Step	Statement	x	y
1	`y = 2`	?	**2**
2	`x = 0`	**0**	2
3	`y = y ^ 2`	0	**4**
4	`If x < 256 Then`	True	
5	`x = x + y`	**4**	
6	`Console.WriteLine(x & ", " & y)`	It displays: 4, 4	
7	`Loop While y < 65535`	True	
8	`y = y ^ 2`	4	**16**
9	`If x < 256 Then`	True	

10	`x = x + y`	**20**	16
11	`Console.WriteLine(x & ", " & y)`	It displays: 20, 16	
12	`Loop While y < 65535`	True	
13	`y = y ^ 2`	20	**256**
14	`If x < 256 Then`	True	
15	`x = x + y`	**276**	256
16	`Console.WriteLine(x & ", " & y)`	It displays: 276, 256	
17	`Do While y < 65535`	True	
18	`y = y ^ 2`	276	**65536**
19	`If x < 256 Then`	False	
20	`Console.WriteLine(x & ", " & y)`	It displays: 276, 65536	
21	`Loop 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 Mod 2 <> 0 Then`	False				
7	`ElseIf d Mod 2 = 0 Then`	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	`Loop While c < 11`	True				
12	`x = a + b`	4	5	0	5	**9**
13	`If x Mod 2 <> 0 Then`	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	`Loop While c < 11`	True				
18	`x = a + b`	5	5	5	5	**10**

19	`If x Mod 2 <> 0 Then`	False				
20	`ElseIf d Mod 2 = 0 Then`	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	`Loop While c < 11`	True				
25	`x = a + b`	5	5	8	5	**10**
26	`If x Mod 2 <> 0 Then`	False				
27	`ElseIf d Mod 2 = 0 Then`	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	`Loop 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

```
Sub Main(args As String())
  Dim a, total As Double
  Dim i, n As Integer

  n = Console.ReadLine()
  total = 0

  i = 1
  Do While i <= n
    a = Console.ReadLine()
    total = total + a
```

```
    i += 1
  Loop

  Console.WriteLine(total)
  If n > 0 Then
    Console.WriteLine(total / n)
  End If
End Sub
```

14. Solution

```
Sub Main(args As String())
  Dim a, i, n, p As Integer
  Dim count As Integer = 0

  n = Console.ReadLine()
  p = 1

  i = 1
  Do While i <= n
    a = Console.ReadLine()
    If a Mod 2 = 0 Then
      p = p * a
      count += 1
    End If
    i += 1
  Loop

  If count > 0 Then
    Console.WriteLine(p)
  Else
    Console.WriteLine("You entered no even integers")
  End If
End Sub
```

15. Solution

```
Sub Main(args As String())
  Dim a, i, total As Integer

  total = 0

  i = 1
  Do While i <= 100
    a = Console.ReadLine()
    If a Mod 10 = 0 Then
      total = total + a
    End If
    i += 1
  Loop
  Console.WriteLine(total)
End Sub
```

16. Solution

```
Sub Main(args As String())
  Dim a, i, total As Integer

  total = 0

  i = 1
  Do While i <= 20
    a = Console.ReadLine()
    If a >= 100 And a <= 999 Then
      total = total + a
    End If
    i += 1
  Loop
  Console.WriteLine(total)
End Sub
```

17. Solution

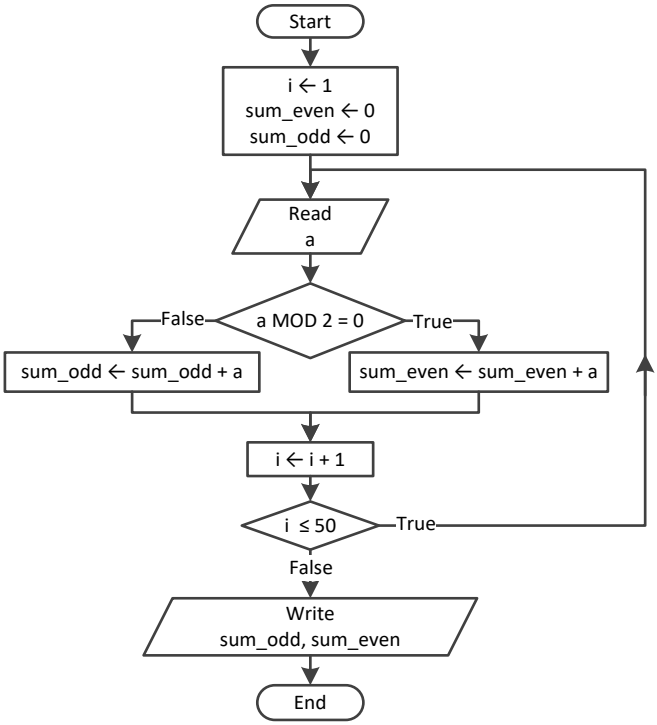
```
Sub Main(args As String())
  Dim a, p As Double

  p = 1

  a = Console.ReadLine()
  Do While a <> 0
    p = p * a
    a = Console.ReadLine()
  Loop
  Console.WriteLine(p)
End Sub
```

Step	Statement	a	p
1	p = 1	?	**1.0**
2	a = Console.ReadLine()	**3.0**	1.0
3	Do While a <> 0	True	
4	p = p * a	3.0	**3.0**
5	a = Console.ReadLine()	**2.0**	3.0
6	Do While a <> 0	True	
7	p = p * a	2.0	**6.0**
8	a = Console.ReadLine()	**9.0**	6.0
9	Do While a <> 0	True	
10	p = p * a	9.0	**54.0**
11	a = Console.ReadLine()	**0.0**	54.0
12	Do While a <> 0	False	
13	Console.WriteLine(p)	It displays: 54	

18. Solution

```
Sub Main(args As String())
  Dim years As Integer
  Dim population As Double

  population = 30000

  years = 0
  Do While population <= 100000
    population += population * 0.03
    years += 1
  Loop
  Console.WriteLine(years)
End Sub
```

19. Solution




```
Sub Main(args As String())
  Dim a, i, sum_even, sum_odd As Integer

  i = 1
  sum_even = 0
  sum_odd = 0
  Do
    a = Console.ReadLine()
    If a Mod 2 = 0 Then
      sum_even += a
    Else
      sum_odd += a
    End If
```

```
    i += 1
  Loop While i <= 50
  Console.WriteLine(sum_even & " " & sum_odd)
End Sub
```

20. Solution

```
Sub Main(args As String())
  Dim a, i, n, p As Integer

  n = Console.ReadLine()
  i = 1
  p = 1
  Do
    a = Console.ReadLine()
    If a < 0 Then
      p *= a
    End If
    i += 1
  Loop While i <= n
  Console.WriteLine(Math.Abs(p))
End Sub
```

21. Solution

```
Sub Main(args As String())
  Dim a, i, p As Integer

  i = 1
  p = 1
  Do
    Console.Write("Enter an integer: ")
    a = Console.ReadLine()
    If a >= 500 And a <= 599 Then
      p *= a
    End If
    i += 1
  Loop While i <= 5
  Console.WriteLine(p)
End Sub
```

22. Solution

```
Sub Main(args As String())
  Dim population As Double
  Dim years As Integer

  population = 50000

  years = 0
  Do
    population -= population * 0.10
    years += 1
```

```
    Loop While population >= 20000
    Console.WriteLine(years)
End Sub
```

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. d
14. b

26.5 Review Exercises

1. Solution

Step	Statement	a	b	j
1	`a = 0`	**0**	?	?
2	`b = 0`	0	**0**	?
3	`j = 0`	0	0	**0**
4	`j <= 8`		True	
5	`If j < 5 Then`		True	
6	`b += 1`	0	**1**	0
7	`j += 2`	0	1	**2**
8	`j <= 8`		True	
9	`If j < 5 Then`		True	
10	`b += 1`	0	**2**	2
11	`j += 2`	0	2	**4**
12	`j <= 8`		True	
13	`If j < 5 Then`		True	
14	`b += 1`	0	**3**	4
15	`j += 2`	0	3	**6**
16	`j <= 8`		True	
17	`If j < 5 Then`		False	
18	`a += j - 1`	**5**	3	6

19	`j += 2`	5	3	**8**
20	`j <= 8`	True		
21	`If j < 5 Then`	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 = Console.ReadLine()`	**10**	?	?
2	`b = a`	10	**10**	?
3	`j = a - 5`	10	10	**5**
4	`j <= a`	True		
5	`If j Mod 2 <> 0 Then`	True		
6	`b = a + j + 5`	10	**20**	5
7	`j += 2`	10	20	**7**
8	`j <= a`	True		
9	`If j Mod 2 <> 0 Then`	True		
10	`b = a + j + 5`	10	**22**	7
11	`j += 2`	10	22	**9**
12	`j <= a`	True		
13	`If j Mod 2 <> 0 Then`	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 = Console.ReadLine()`	**21**	?	?
2	`b = a`	21	**21**	?
3	`j = a - 5`	21	21	**16**
4	`j <= a`	True		
5	`If j Mod 2 <> 0 Then`	False		
6	`b = a + j + 5`	21	**5**	16
7	`j += 2`	21	5	**18**

8	`j <= a`	True		
9	`If j Mod 2 <> 0 Then`	False		
10	`b = a + j + 5`	21	**3**	18
11	`j += 2`	21	3	**20**
12	`j <= a`	True		
13	`If j Mod 2 <> 0 Then`	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 = 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 Or x > 30 Then`	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 Or x > 30 Then`	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 Or x > 30 Then`	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 Or x > 30 Then`	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			

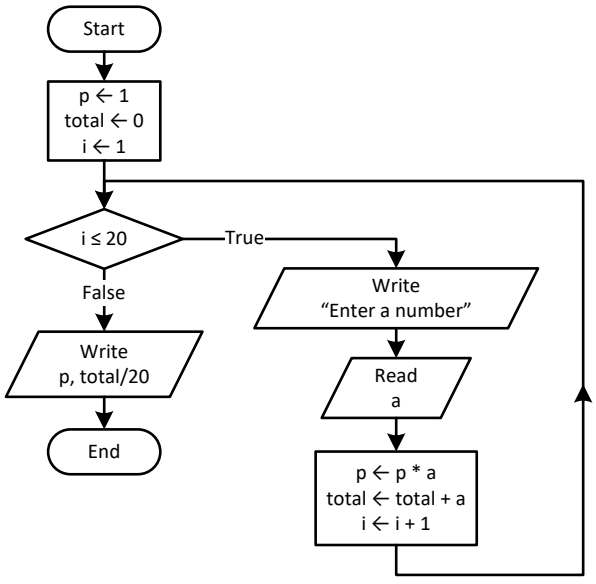
4. Solution

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




```
Sub Main(args As String())
  Dim a, p, total As Double
  Dim i As Integer

  p = 1
```

```
  total = 0
  For i = 1 To 20
    Console.Write("Enter a number: ")
    a = Console.ReadLine()
    p = p * a
    total = total + a
  Next
  Console.WriteLine(p)
  Console.WriteLine(total / 20)
End Sub
```

7. Solution

```
Sub Main(args As String())
  Dim i As Double

  For i = 0 To 360 Step 0.5
    Console.WriteLine(Math.Sin(i * Math.PI / 180))
  Next
End Sub
```

8. Solution

```
Sub Main(args As String())
  Dim deg, i As Integer

  Console.Write("Enter degrees: ")
  deg = Console.ReadLine()
  For i = 0 To deg
    Console.WriteLine(Math.Cos(i * Math.PI / 180))
  Next
End Sub
```

9. Solution

```
Sub Main(args As String())
  Dim i, s As Integer

  s = 0
  For i = 1 To 99 Step 2
    s += i
  Next
  Console.WriteLine(s)
End Sub
```

10. Solution

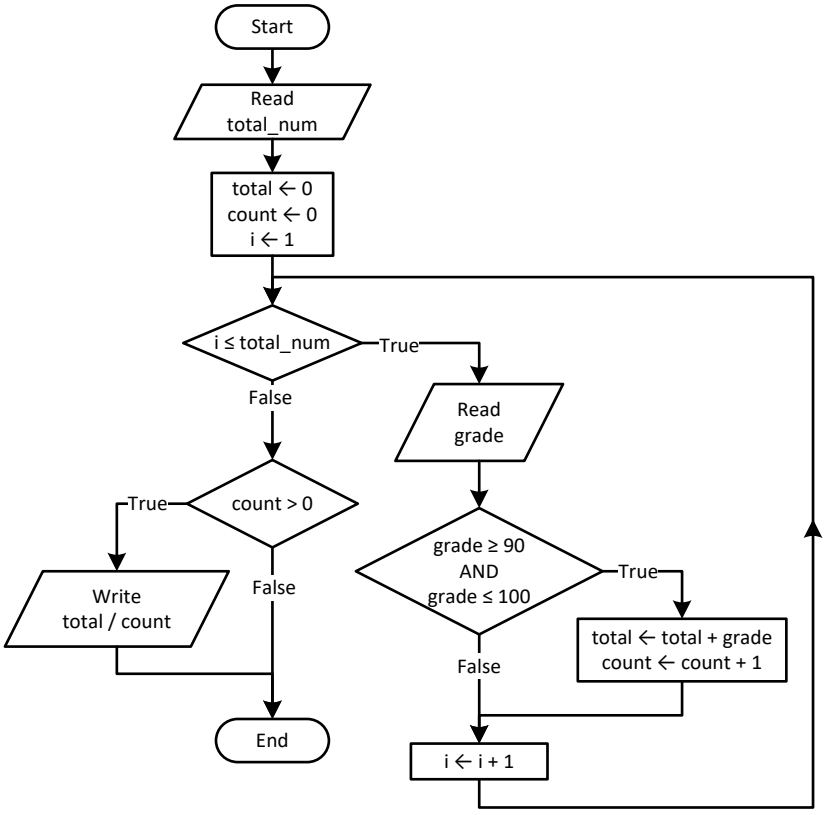
```
Sub Main(args As String())
  Dim i, n As Integer
  Dim p As Double

  n = Console.ReadLine()
  p = 1
```

```
  For i = 2 To 2 * n Step 2
    p *= i ^ (i - 1)
  Next
  Console.WriteLine(p)
End Sub
```

11. Solution

```
Sub Main(args As String())
  Dim i, offset, s As Integer

  s = 0
  i = 1
  offset = 0
  Do While i <= 191
    s += i
    offset += 1
    i += offset
  Loop
  Console.WriteLine(s)
End Sub
```

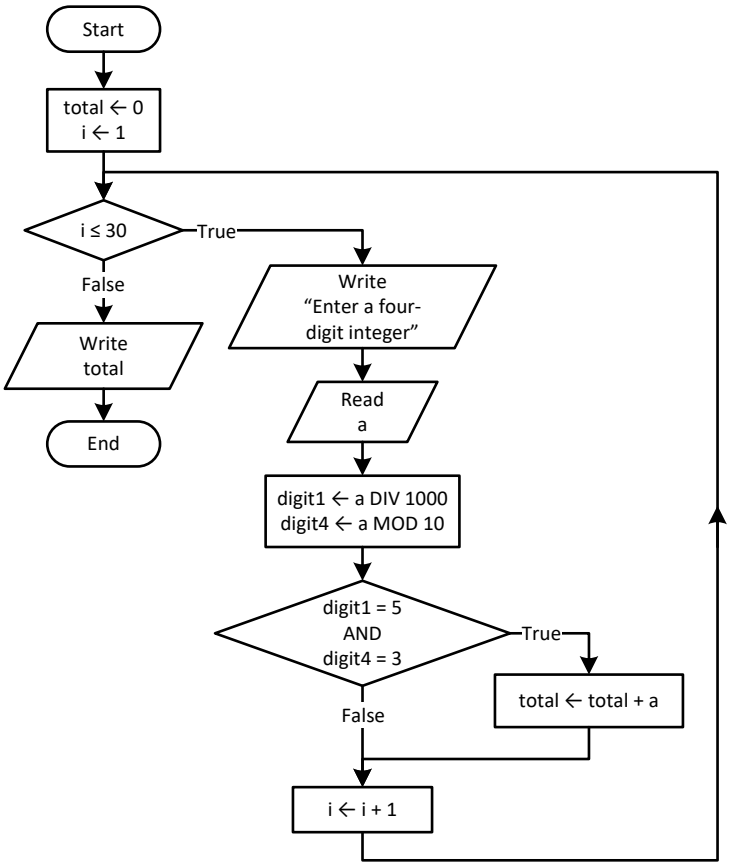
12. Solution




```
Sub Main(args As String())
  Dim count, grade, i, total_num, total As Integer

  total_num = Console.ReadLine()
```

```
  total = 0
  count = 0
  For i = 1 To total_num
    grade = Console.ReadLine()
    If grade >= 90 And grade <= 100 Then
      total += grade
      count += 1
    End If
  Next
  If count > 0 Then
    Console.WriteLine(total / count)
  End If
End Sub
```

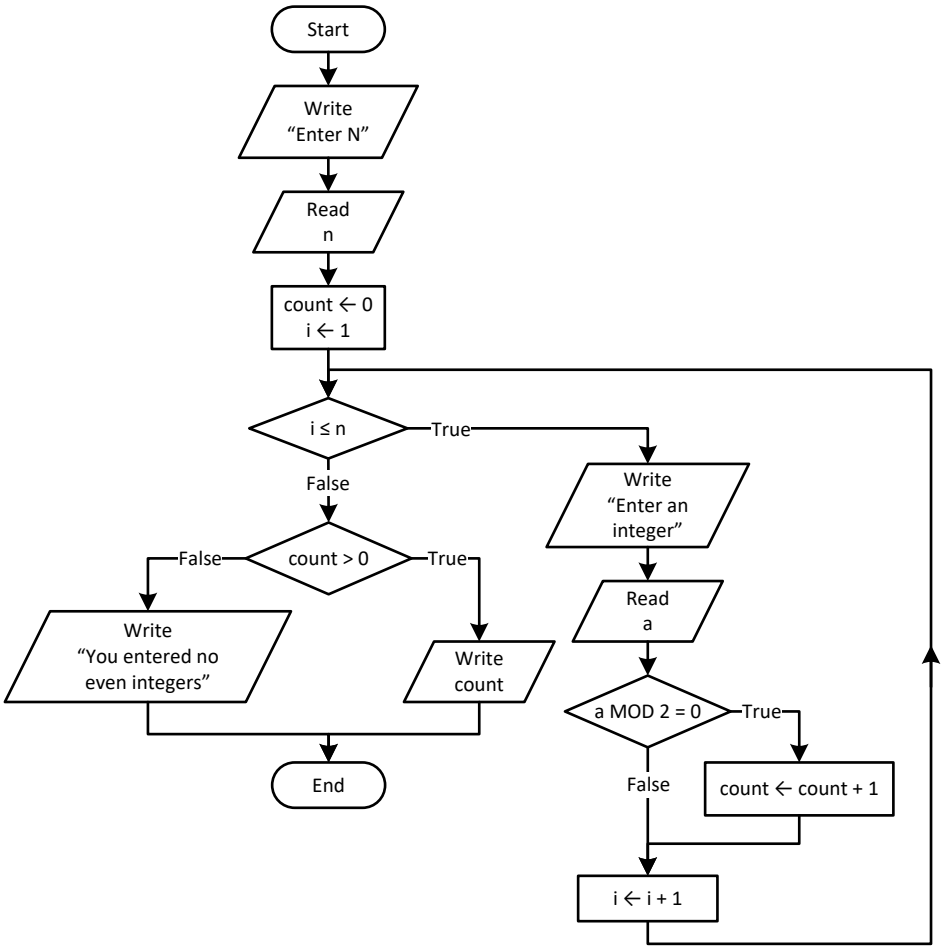
13. Solution




```
Sub Main(args As String())
  Dim a, digit1, digit4, i, total As Integer

  total = 0
  For i = 1 To 30
    Console.Write("Enter a four-digit integer: ")
    a = Console.ReadLine()
    digit1 = a \ 1000
    digit4 = a Mod 10
    If digit1 = 5 And digit4 = 3 Then
```

```
      total += a
    End If
  Next
  Console.WriteLine(total)
End Sub
```

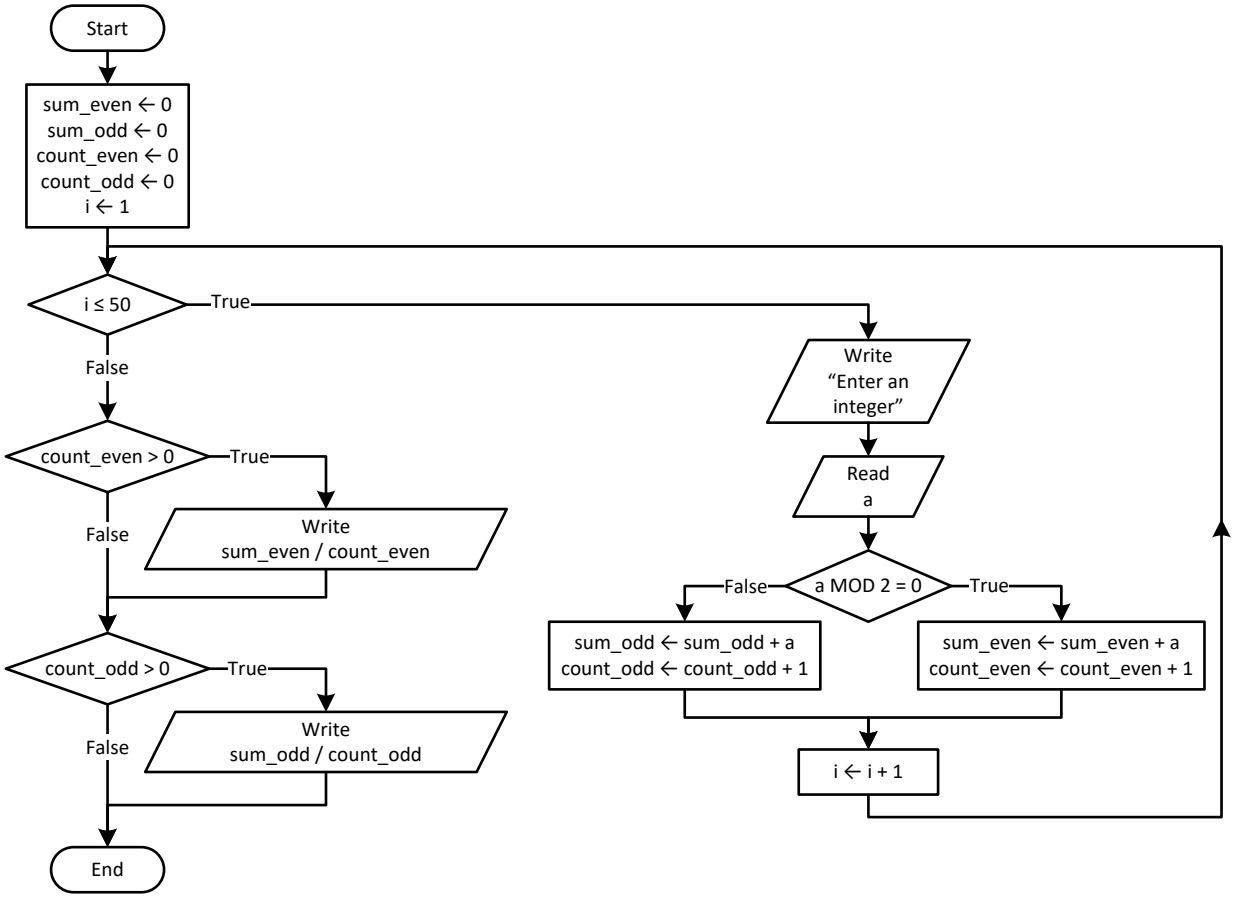
14. Solution




```
Sub Main(args As String())
  Dim a, count, i, n As Integer

  Console.Write("Enter N: ")
  n = Console.ReadLine()
  count = 0
  For i = 1 To n
    Console.Write("Enter an integer: ")
    a = Console.ReadLine()
    If a Mod 2 = 0 Then
      count += 1
    End If
  Next
  If count > 0 Then
    Console.WriteLine(count)
```

```
  Else
    Console.WriteLine("You entered no even integers")
  End If
End Sub
```

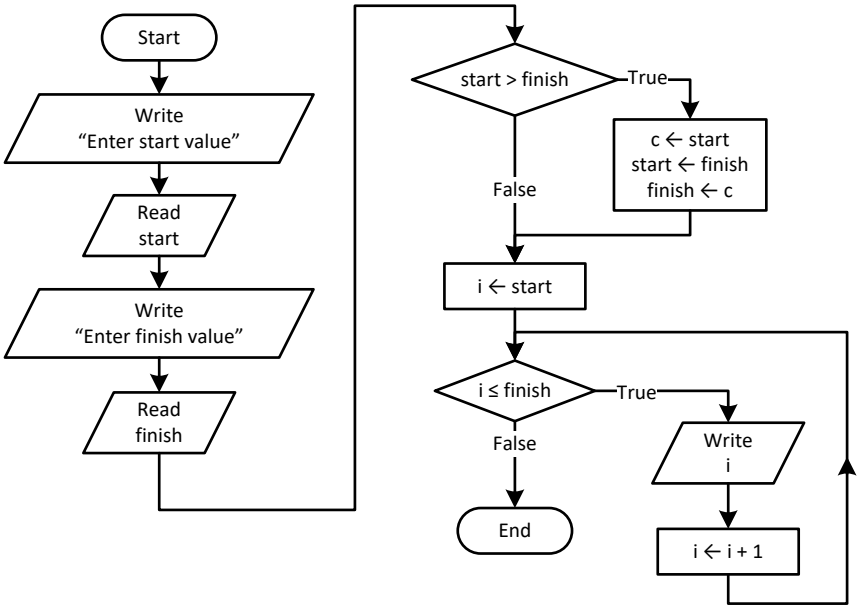
15. Solution




```
Sub Main(args As String())
  Dim a, count_even, count_odd, i, sum_even, sum_odd As Integer

  sum_even = 0
  sum_odd = 0
  count_even = 0
  count_odd = 0
  For i = 1 To 50
    Console.Write("Enter an integer: ")
    a = Console.ReadLine()
    If a Mod 2 = 0 Then
      sum_even += a
      count_even += 1
    Else
      sum_odd += a
      count_odd += 1
    End If
  Next
```

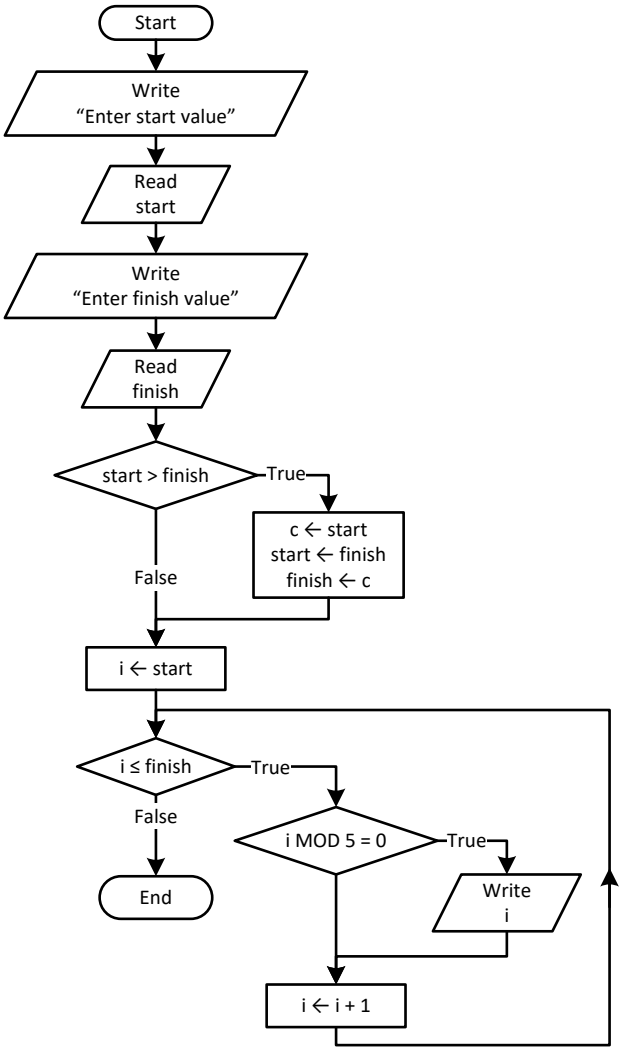
```
  If count_even > 0 Then
    Console.WriteLine(sum_even / count_even)
  End If
  If count_odd > 0 Then
    Console.WriteLine(sum_odd / count_odd)
  End If
End Sub
```

16. Solution




```
Sub Main(args As String())
  Dim c, finish, i, start As Integer

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

  If start > finish Then
    c = start
    start = finish
    finish = c
  End If

  For i = start To finish
    Console.WriteLine(i)
  Next
End Sub
```

17. Solution




```
Sub Main(args As String())
  Dim c, finish, i, start As Integer

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

  If start > finish Then
    c = start
    start = finish
    finish = c
  End If

  For i = start To finish
    If i Mod 5 = 0 Then
      Console.WriteLine(i)
    End If
```

```
  Next
End Sub
```

18. Solution

First approach

```
Sub Main(args As String())
  Dim exp, i As Integer
  Dim p, b As Double

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

  p = 1
  If exp >= 0 Then
    For i = 1 To exp
      p *= b
    Next
  Else
    For i = 1 To -exp
      p *= 1 / b
    Next
  End If
  Console.WriteLine(p)
End Sub
```

Second approach

```
Sub Main(args As String())
  Dim exp, i As Integer
  Dim p, b As Double

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

  p = 1
  For i = 1 To Math.Abs(exp)
    p *= b
  Next

  If exp < 0 Then
    p = 1 / p
  End If
  Console.WriteLine(p)
End Sub
```

19. Solution

```
Sub Main(args As String())
  Dim count, i, words As Integer
```

```
  Dim msg, character As String

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

  count = 0
  For i = 0 To msg.Length - 1
    character = msg(i)
    If character = " " Then
      count += 1
    End If
  Next
  words = count + 1

  Console.WriteLine("The message entered contains " & words & " words")
End Sub
```

20. Solution

```
Sub Main(args As String())
  Dim characters, count, i, words As Integer
  Dim msg, character As String

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

  characters = msg.Length
  count = 0
  For i = 0 To characters - 1
    character = msg(i)
    If character = " " Then
      count += 1
    End If
  Next

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

21. Solution

```
Sub Main(args As String())
  Dim message As String
  Dim character As Char
  Dim consonants As String = "BCDFGHJKLMNPQRSTVWZYX"
  Dim i, count As Integer

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

  count = 0
  For i = 0 To message.Length - 1
```

```
    character = message(i)

    If consonants.IndexOf(character) <> -1 Then 'If character is found in consonants
      count += 1
    End If
  Next
  Console.WriteLine("Consonants: " & count)
End Sub
```

22. Solution

```
Sub Main(args As String())
  Dim message As String
  Dim character As Char
  Dim vowels As String = "AEIOU"
  Dim consonants As String = "BCDFGHJKLMNPQRSTVWZYX"
  Dim digits As String = "0123456789"
  Dim i, countv, countc, countd As Integer

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

  countv = 0
  countc = 0
  countd = 0
  For i = 0 To message.Length - 1
    character = message(i)

    If vowels.IndexOf(character) <> -1 Then 'If character is found in vowels
      countv += 1
    ElseIf consonants.IndexOf(character) <> -1 Then 'If character is found in consonants
      countc += 1
    ElseIf digits.IndexOf(character) <> -1 Then 'If character is found in digits
      countd += 1
    End If
  Next
  Console.WriteLine("Vowels: " & countv)
  Console.WriteLine("Consonants: " & countc)
  Console.WriteLine("Digits: " & countd)
End Sub
```

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

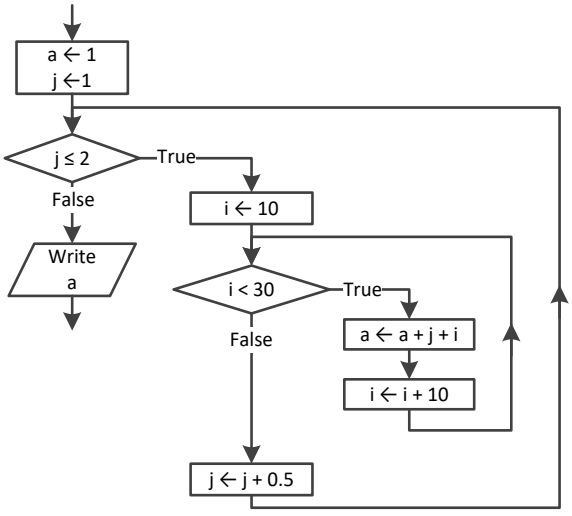
1. b
2. a
3. c
4. a
5. b

27.5 Review Exercises

1. Solution

i. 10
ii. A value greater than or equal to 4.5 and less than 5.0
iii. −7 (or −6)
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 -= 1`	3	1	**2**
8	`j >= i`		True	
9	`s = s + i * j`	**5**	1	2
10	`j -= 1`	5	1	**1**
11	`j >= i`		True	
12	`s = s + i * j`	**6**	1	1
13	`j -= 1`	6	1	**0**
14	`j >= i`		False	
15	`i += 1`	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 -= 1`	12	2	**2**
21	`j >= i`		True	
22	`s = s + i * j`	**16**	2	2
23	`j -= 1`	16	2	**1**
24	`j >= i`		False	
25	`i += 1`	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 -= 1`	25	3	**2**
31	`j >= i`		False	
32	`i += 1`	25	**4**	2
33	`i <= 4`		True	
34	`j = 3`	25	4	**3**
35	`j >= i`		False	
36	`i += 1`	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 += 1	26	20	**2**	?
8	i <= 3	True			
9	s = s + y	**46**	20	2	?
10	y -= 5	46	**15**	2	?
11	i += 1	46	15	**3**	?
12	i <= 3	True			
13	s = s + y	**61**	15	3	?
14	y -= 5	61	**10**	3	?
15	i += 1	61	10	**4**	?
16	i <= 3	False			
17	ans = Console.ReadLine()	61	10	4	**"NO"**
18	Loop 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 += 1	26	20	**2**	?
8	i <= 3	True			
9	s = s + y	**46**	20	2	?
10	y -= 5	46	**15**	2	?
11	i += 1	46	15	**3**	?
12	i <= 3	True			
13	s = s + y	**61**	15	3	?
14	y -= 5	61	**10**	3	?
15	i += 1	61	10	**4**	?

16	`i <= 3`	False			
17	`ans = Console.ReadLine()`	61	10	4	**"YES"**
18	`Loop 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 += 1`	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 += 1`	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 += 1`	76	−5	**4**	"YES"
32	`i <= 3`	False			
33	`ans = Console.ReadLine()`	76	−5	4	**"NO"**
34	`Loop 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 += 1`	26	20	**2**	?
8	`i <= 3`	True			
9	`s = s + y`	**46**	20	2	?
10	`y -= 5`	46	**15**	2	?
11	`i += 1`	46	15	**3**	?
12	`i <= 3`	True			
13	`s = s + y`	**61**	15	3	?
14	`y -= 5`	61	**10**	3	?
15	`i += 1`	61	10	**4**	?

16	`i <= 3`	False			
17	`ans = Console.ReadLine()`	61	10	4	**"YES"**
18	`Loop 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 += 1`	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 += 1`	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 += 1`	76	−5	**4**	"YES"
32	`i <= 3`	False			
33	`ans = Console.ReadLine()`	76	−5	4	**"YES"**
34	`Loop 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 += 1`	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 += 1`	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 += 1`	46	−20	**4**	"YES"
48	`i <= 3`	False			
49	`ans = Console.ReadLine()`	46	−20	4	**"NO"**
50	`Loop While ans = "YES"`	False			
51	`Console.WriteLine(s)`	It displays: 46			

5. Solution

```
Sub Main(args As String())
  Dim hour, minutes As Integer

  For hour = 0 To 23
    For minutes = 0 To 59
      Console.WriteLine(hour & vbTab & minutes)
    Next
  Next
End Sub
```

6. Solution

```
Sub Main(args As String())
  Dim i, j As Integer

  For i = 5 To 1 Step -1
    For j = 1 To i
      Console.Write(i & " ")
    Next
    Console.WriteLine()
  Next
End Sub
```

7. Solution

```
Sub Main(args As String())
  Dim i, j As Integer

  For i = 0 To 5
    For j = 0 To i
      Console.Write(j & " ")
    Next
    Console.WriteLine()
  Next
End Sub
```

8. Solution

```
Sub Main(args As String())
  Dim i, j As Integer

  For i = 1 To 4
    For j = 1 To 10
      Console.Write("* ")
    Next
    Console.WriteLine()
  Next
End Sub
```

9. Solution

```
Sub Main(args As String())
  Dim i, j, y As Integer

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

  For i = 1 To y
    For j = 1 To y
      Console.Write("* ")
    Next
    Console.WriteLine()
  Next
End Sub
```

10. Solution

```
Sub Main(args As String())
  Dim i, j, y As Integer

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

  For j = 1 To y
    Console.Write("* ")
  Next
  Console.WriteLine()

  For i = 1 To y - 2
    Console.Write("* ")
    For j = 1 To y - 2
      Console.Write("  ")
    Next
    Console.WriteLine("* ")
  Next

  For j = 1 To y
    Console.Write("* ")
  Next
End Sub
```

11. Solution

```
Sub Main(args As String())
  Dim i, j As Integer

  For i = 1 To 5
    For j = 1 To i
      Console.Write("* ")
    Next
    Console.WriteLine()
  Next
```

```
  For i = 4 To 1 Step -1
    For j = 1 To i
      Console.Write("* ")
    Next
    Console.WriteLine()
  Next
End Sub
```

Chapter 28

28.8 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

28.9 Review Questions: Multiple Choice

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

28.10 Review Exercises

1. Solution

```
count_names = 0
count_not_johns = 0
name = ""
Console.Write("Enter a name: ")
name = Console.ReadLine()
Do While name <> "STOP"
  Console.Write("Enter a name: ")
  name = Console.ReadLine()
  count_names += 1
  If name <> "John" Then
    count_not_johns += 1
  End If
  Console.Write("Enter a name: ")
  name = Console.ReadLine()
Loop
Console.WriteLine(count_names & " names entered")
Console.WriteLine("Names other than John entered " & count_not_johns & " times")
```

2. Solution

First approach

```
Sub Main(args As String())
  Dim text, character As String
  Dim found As Boolean
  Dim i As Integer

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

  found = False
  For i = 0 To text.Length - 1
```

```
    character = text(i)
    If character = " " Then
      found = True
      Exit For
    End If
  Next

  If Not found Then
    Console.WriteLine("One Single Word")
  Else
    Console.WriteLine("Complete Sentence")
  End If
End Sub
```

Second approach

```
Sub Main(args As String())
  Dim text As String

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

  If text.IndexOf(" ") = -1 Then
    Console.WriteLine("One Single Word")
  Else
    Console.WriteLine("Complete Sentence")
  End If
End Sub
```

3. Solution

First approach

```
Sub Main(args As String())
  Dim sentence, character As String
  Dim found As Boolean
  Dim i As Integer
  Dim digits As String = "0123456789"

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

  found = False
  For i = 0 To sentence.Length - 1
    character = sentence(i)
    If digits.IndexOf(character) <> -1 Then
      found = True
      Exit For
    End If
  Next

  If found Then
    Console.WriteLine("The sentence contains a number")
  End If
End Sub
```

Second approach

```
Sub Main(args As String())
  Dim sentence As String
  Dim found As Boolean
  Dim i As Integer
  Dim digit As String

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

  found = False
  For i = 0 To 9
    digit = i.ToString()
    If sentence.IndexOf(digit) <> -1 Then
      found = True
      Exit For
    End If
  Next

  If found Then
    Console.WriteLine("The sentence contains a number")
  End If
End Sub
```

4. Solution

```
Console.WriteLine("Printing all integers from 1 to 100")
i = 1
Do While i < 101
  Console.WriteLine(i)
  i += 1
Loop
```

5. Solution

```
Console.WriteLine("Printing odd integers from 1 to 99")
i = 1
Do While i < 100
  Console.WriteLine(i)
  i += 2
Loop
```

6. Solution

```
s = 0
For i = 1 To 100
  number = Console.ReadLine()
  s = s + number
Next
average = s / 100.0
Console.WriteLine(average)
```

7. Solution

```
Dim i, denom As Integer
Dim s As Double

s = 0

denom = 1
For i = 1 To 100
  denom *= i
Next

For i = 1 To 100
  s += i / denom
Next
Console.WriteLine(s)
```

8. Solution

```
Sub Main(args As String())
  Dim i, j As Integer

  For i = 1 To 4
    For j = 1 To 4
      Console.WriteLine(i & " x " & j & " = " & (i * j))
    Next
  Next
End Sub
```

9. Solution

```
Sub Main(args As String())
  Dim i, j As Integer

  Console.Write(vbTab & "|" & vbTab)
  For i = 1 To 12
    Console.Write(i & vbTab)
  Next
  Console.WriteLine()

  For i = 1 To 12
    Console.Write("---------")
  Next
  Console.WriteLine()

  For i = 1 To 12
    Console.Write(i & vbTab & "|" & vbTab)
    For j = 1 To 12
      Console.Write(i * j & vbTab)
    Next
    Console.WriteLine()
  Next
End Sub
```

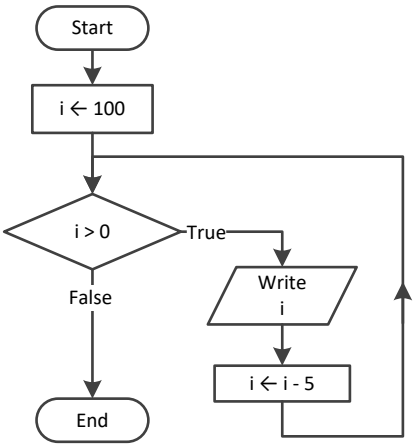
10. Solution

```
Sub Main(args As String())
  Dim i, j, n As Integer

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

  Console.Write(vbTab & "|" & vbTab)
  For i = 1 To n
    Console.Write(i & vbTab)
  Next
  Console.WriteLine()

  For i = 1 To n
    Console.Write("---------")
  Next
  Console.WriteLine()

  For i = 1 To n
    Console.Write(i & vbTab & "|" & vbTab)
    For j = 1 To n
      Console.Write(i * j & vbTab)
    Next
    Console.WriteLine()
  Next
End Sub
```

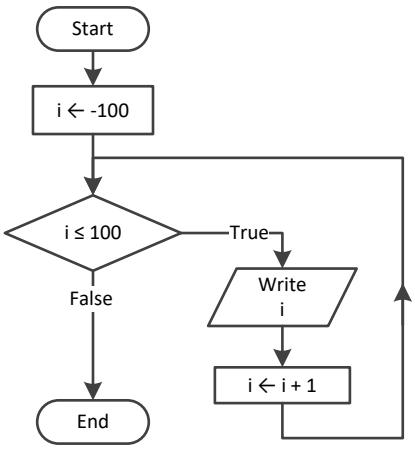
Chapter 29

29.4 Review Exercises

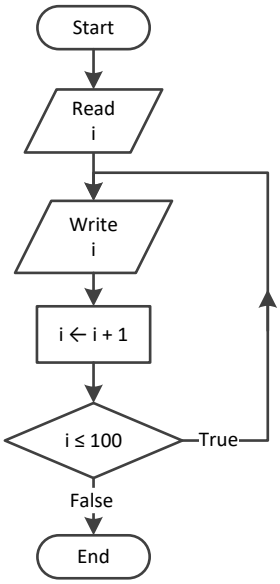
1. Solution



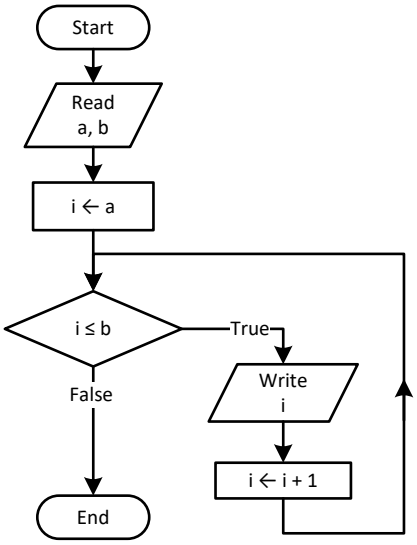
2. Solution



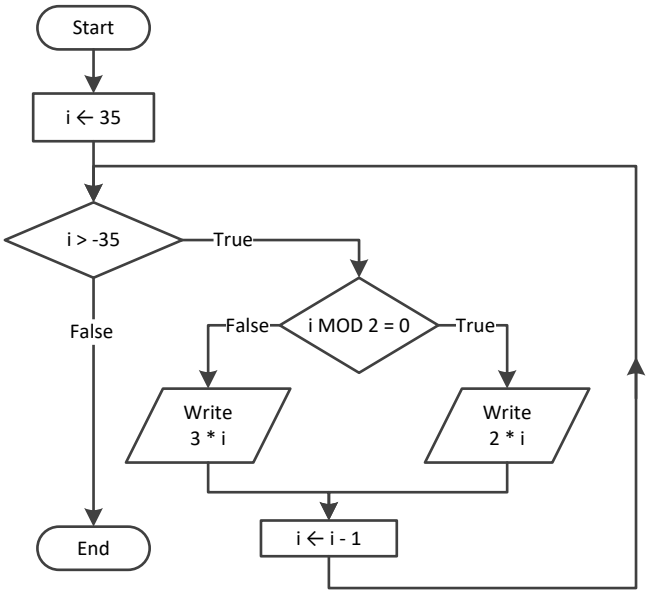
3. Solution



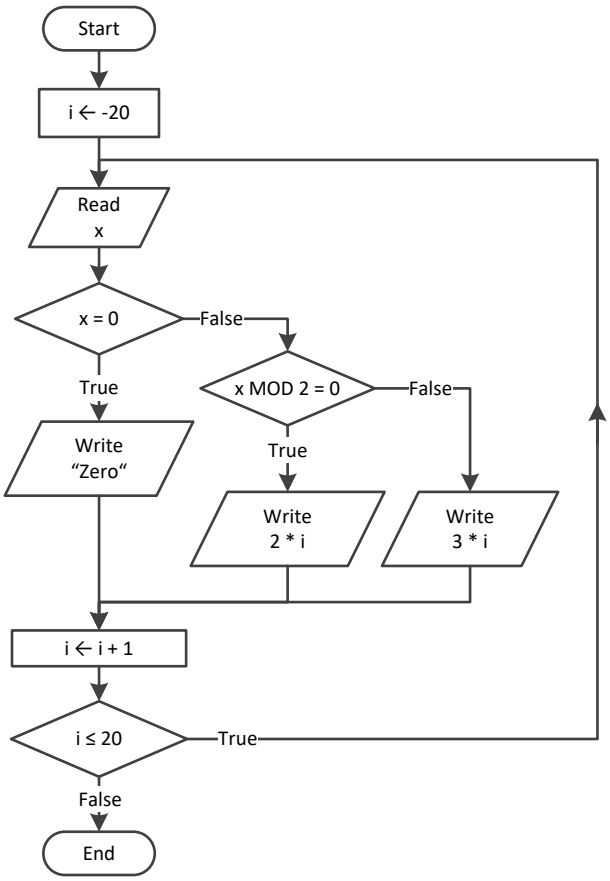
4. Solution



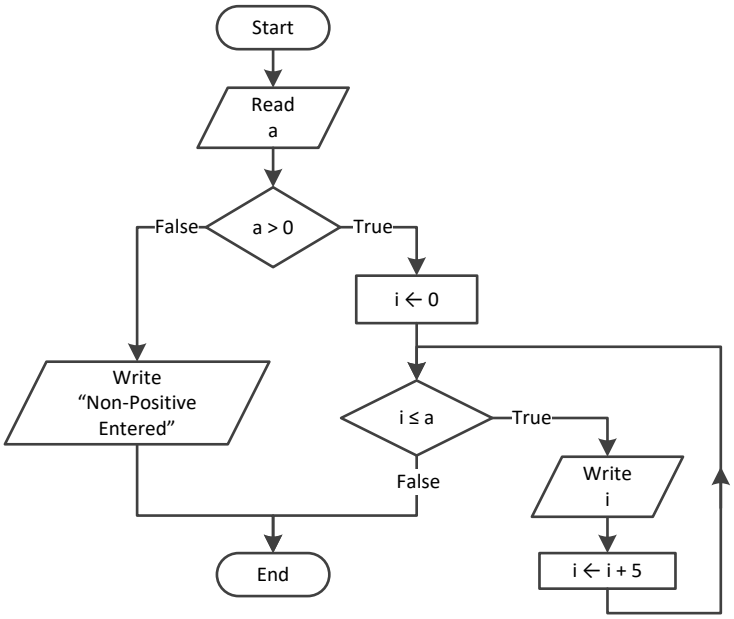
5. Solution



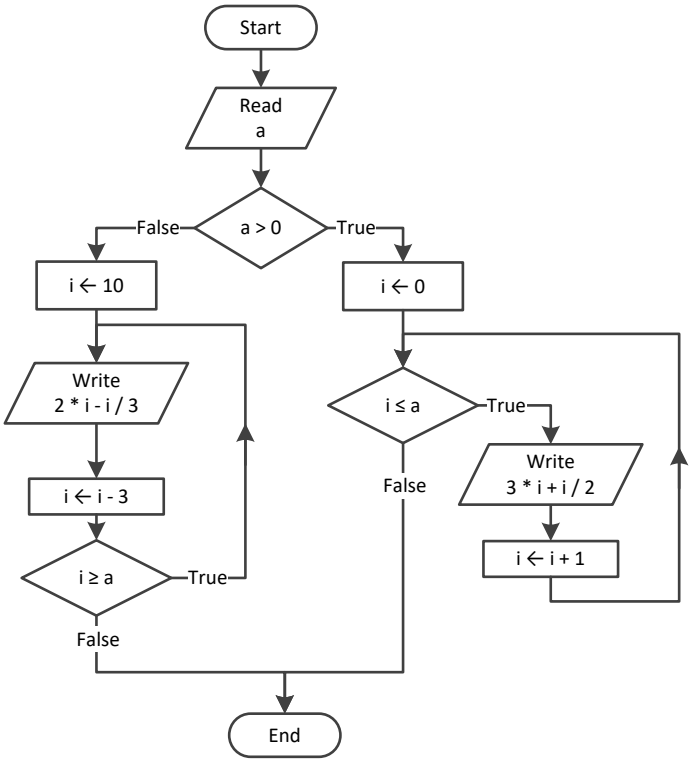
6. Solution



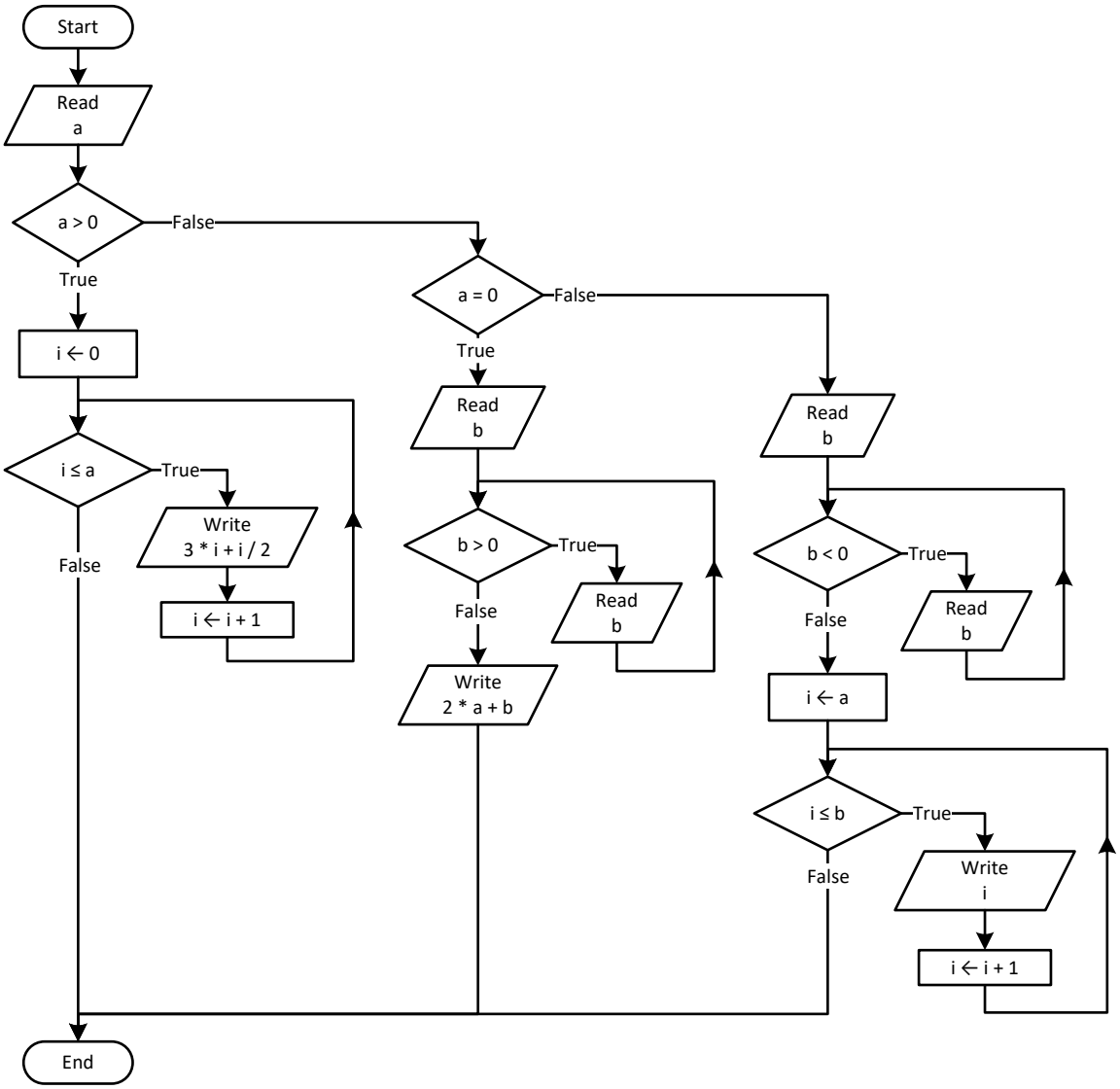
7. Solution



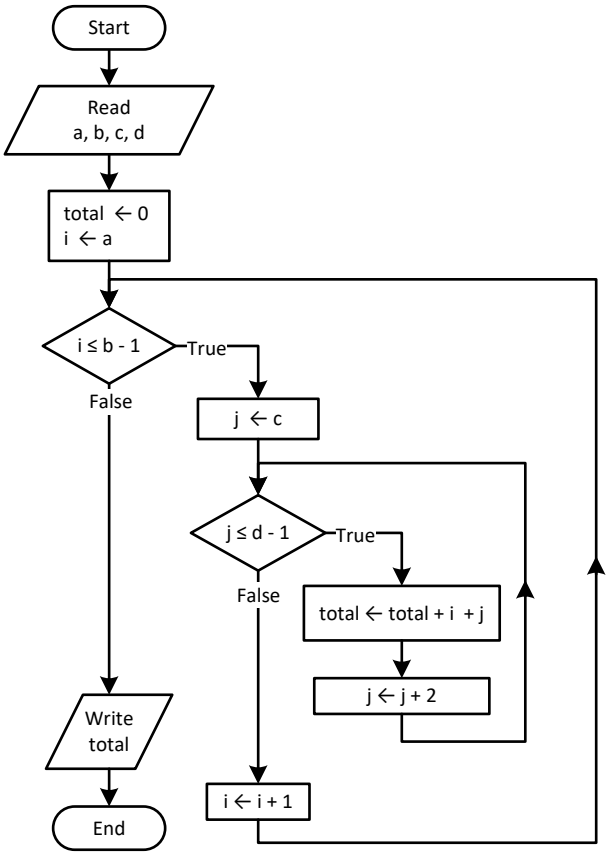
8. Solution



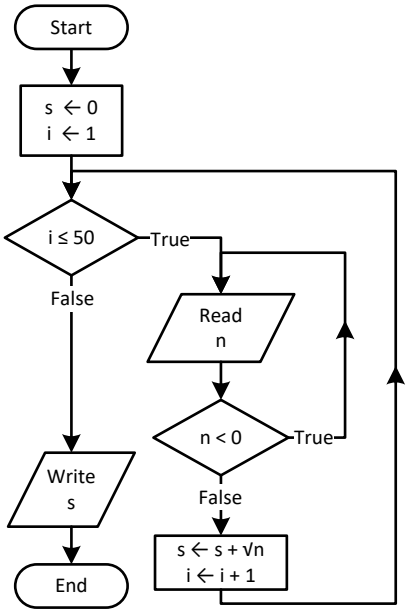
9. Solution


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

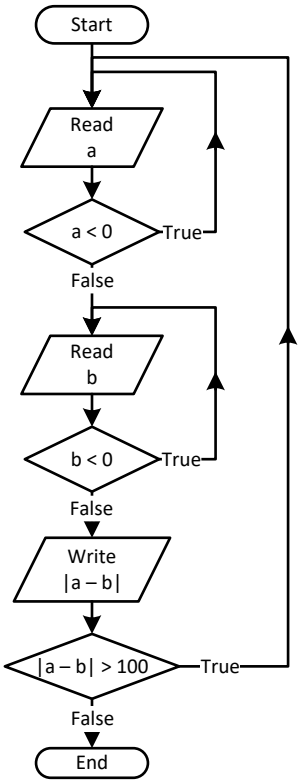
10. Solution



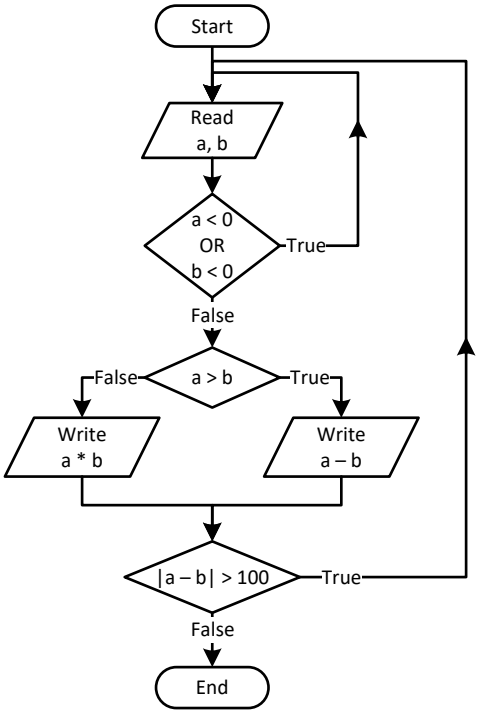
11. Solution



12. Solution



13. Solution



14. Solution

```
i = 1
Do
  Console.WriteLine(i)
  i += 5
Loop While i <= 500
Console.WriteLine("The End")
```

15. Solution

```
Sub Main(args As String())
  Dim a, i As Integer

  i = 0
  a = Console.ReadLine()
  Do
    If i Mod 2 <> 0 Then
      Console.WriteLine(i)
    End If
    i += 5
  Loop While i < a
End Sub
```

16. Solution

```
Sub Main(args As String())
  Dim a, b, i As Integer

  a = Console.ReadLine()
  Do While a <> -1
    Do
      b = Console.ReadLine()
    Loop While b <= a
    For i = a To b
      Console.WriteLine(i)
    Next
    a = Console.ReadLine()
  Loop
End Sub
```

17. Solution

```
Sub Main(args As String())
  Dim i As Integer
  Dim P, S, a As Double

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

```
  Do While True
    If i < 45 Then
      S += a
    Else
      P *= a
    End If
    i += 1
    If i >= 90 Then Exit Do
    a = Console.ReadLine()
  Loop

  Console.WriteLine(S & " " & P)
End Sub
```

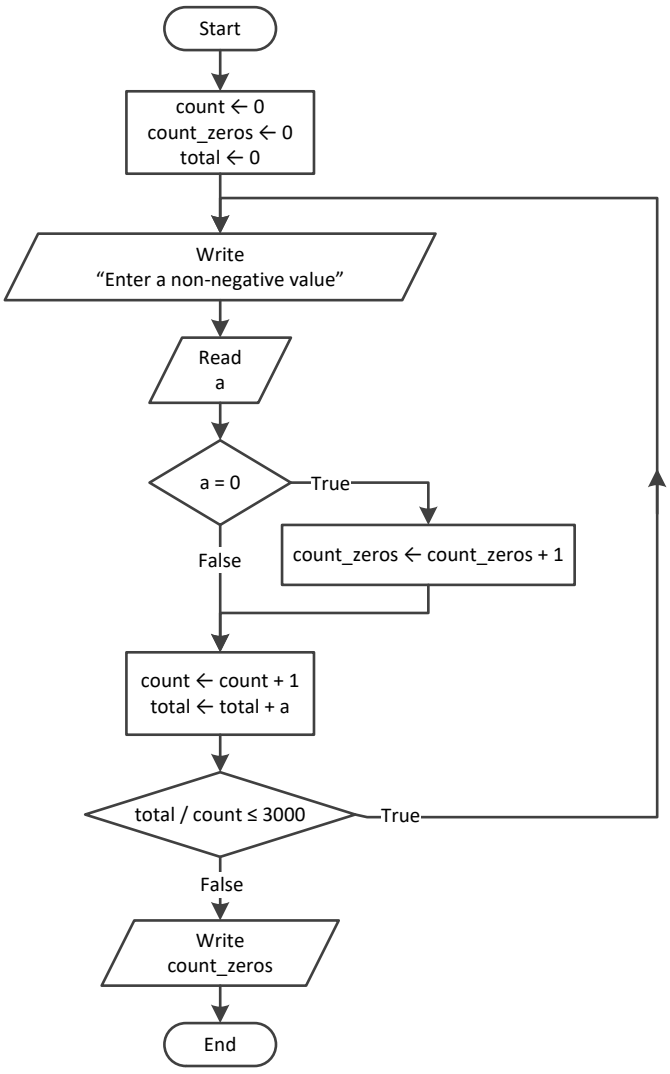
Chapter 30

30.7 Review Questions: True/False

1. True
2. False
3. False
4. True
5. False
6. False
7. False
8. False
9. True

30.8 Review Exercises

1. Solution




```
Sub Main(args As String())
  Dim count, count_zeros As Integer
  Dim a, total As Double

  count = 0
  count_zeros = 0
```

```
  total = 0
  Do
    Console.Write("Enter a non-negative value: ")
    a = Console.ReadLine()
    If a = 0 Then
      count_zeros += 1
    End If
    count += 1
    total += a
  Loop While total / count <= 3000
  Console.WriteLine(count_zeros)
End Sub
```

2. Solution

First approach

```
Sub Main(args As String())
  Dim a, d1, d2, d3, d4, i, r As Integer

  Console.Write("Enter an integer between 1 and 20: ")
  a = Console.ReadLine()
  For i = 1000 To 9999
    d4 = i Mod 10
    r = i \ 10
    d3 = r Mod 10
    r = r \ 10
    d2 = r Mod 10
    d1 = r \ 10
    If d1 + d2 + d3 + d4 < a Then
      Console.WriteLine(i)
    End If
  Next
End Sub
```

Second approach

```
Sub Main(args As String())
  Dim a, d1, d2, d3, d4 As Integer

  Console.Write("Enter an integer between 1 and 20: ")
  a = Console.ReadLine()
  For d1 = 1 To 9
    For d2 = 0 To 9
      For d3 = 0 To 9
        For d4 = 0 To 9
          If d1 + d2 + d3 + d4 < a Then
            Console.WriteLine(d1 * 1000 + d2 * 100 + d3 * 10 + d4)
          End If
        Next
      Next
    Next
  Next
End Sub
```

3. Solution

First approach

```
Sub Main(args As String())
  Dim d1, d2, d3, d4, i, r As Integer

  For i = 1000 To 9999
    d4 = i Mod 10
    r = i \ 10
    d3 = r Mod 10
    r = r \ 10
    d2 = r Mod 10
    d1 = r \ 10
    If d1 > d2 And d2 = d3 And d3 < d4 Then
      Console.WriteLine(i)
    End If
  Next
End Sub
```

Second approach

```
Sub Main(args As String())
  Dim d1, d2, d3, d4 As Integer

  For d1 = 1 To 9
    For d2 = 0 To 9
      For d3 = 0 To 9
        For d4 = 0 To 9
          If d1 > d2 And d2 = d3 And d3 < d4 Then
            Console.WriteLine(d1 * 1000 + d2 * 100 + d3 * 10 + d4)
          End If
        Next
      Next
    Next
  Next
End Sub
```

4. Solution

First approach

```
Sub Main(args As String())
  Dim x, count As Integer

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

  count = 0

  Do While x <> 0
    count += 1
    x = x \ 10
  Loop

  Console.WriteLine(count)
```

```
End Sub
```

Second approach

```
Sub Main(args As String())
  Dim x, count As Integer

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

  'Convert the absolute value of x to string and get its length
  count = Math.Abs(x).ToString().Length

  Console.WriteLine(count)
End Sub
```

5. Solution

```
x = Console.ReadLine()
Do While x <> 1 And x <> 0
  Console.WriteLine("Error")
  x = Console.ReadLine()
Loop
```

6. Solution

```
Do
  gender = Console.ReadLine().ToUpper()
Loop While gender <> "M" And gender <> "F"
```

7. Solution

```
Sub Main(args As String())
  Dim count, x As Integer
  Dim y As Double

  Console.Write("Enter a non-negative number: ")
  x = Console.ReadLine()
  count = 0
  Do While x < 0
    count += 1
    If count = 2 Then Exit Do

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

  If count < 2 Then
    y = Math.Sqrt(x)
    Console.WriteLine(y)
  Else
    Console.WriteLine("Dude, you are dumb!")
  End If
End Sub
```

8. Solution

```
Sub Main(args As String())
  Dim answer As String
  Dim area, r As Double

  Do
    Console.Write("Enter the length of a radius: ")
    r = Console.ReadLine()
    Do While r <= 0
      Console.Write("Invalid radius. Enter the length of a radius: ")
      r = Console.ReadLine()
    Loop

    area = Math.PI * r ^ 2
    Console.WriteLine("The area is: " & area)

    Console.Write("Would you like to repeat? ")
    answer = Console.ReadLine()
  Loop While answer.ToUpper() = "YES"
End Sub
```

9. Solution

```
Sub Main(args As String())
  Dim x, y As Integer

  For x = -100 To 100
    For y = -100 To 100
      If 5 * x + 3 * y ^ 2 = 0 Then
        Console.WriteLine(x & ", " & y)
      End If
    Next
  Next
End Sub
```

10. Solution

```
Sub Main(args As String())
  Dim x, y, z As Integer

  For x = -10 To 10
    For y = -10 To 10
      For z = -10 To 10
        If (x + y) / 2 + 3 * z ^ 2 / (x + 3 * y + 45) = x / 3 Then
          Console.WriteLine(x & ", " & y & ", " & z)
        End If
      Next
    Next
  Next
End Sub
```

11. Solution

```
Sub Main(args As String())
  Dim m1, m2, m3, s As Integer

  m1 = Console.ReadLine()
  m2 = Console.ReadLine()
  m3 = Console.ReadLine()

  s = 0
  Do While m2 <> 0
    If m2 Mod 2 <> 0 Then
      s += m1
    End If
    m1 *= 2
    m2 = m2 \ 2
  Loop

  m1 = s
  m2 = m3

  s = 0
  Do While m2 <> 0
    If m2 Mod 2 <> 0 Then
      s += m1
    End If
    m1 *= 2
    m2 = m2 \ 2
  Loop

  Console.WriteLine(s)
End Sub
```

12. Solution

```
Sub Main(args As String())
  Dim x, number_of_divisors, i As Integer

  x = Console.ReadLine()
  Do While x <= 0
    Console.WriteLine("Error! You must enter a positive integer")
    x = Console.ReadLine()
  Loop

  number_of_divisors = 2
  For i = 2 To x \ 2
    If x Mod i = 0 Then
      number_of_divisors += 1
    End If
  Next
  Console.WriteLine(number_of_divisors)
End Sub
```

13. Solution

```
Sub Main(args As String())
  Dim x, number_of_divisors, i As Integer

  Console.Write("Enter an integer greater than 1: ")
  x = Console.ReadLine()
  Do While x <= 1
    Console.WriteLine("Error!")
    x = Console.ReadLine()
  Loop

  number_of_divisors = 2
  For i = 2 To x \ 2
    If x Mod i = 0 Then
      number_of_divisors += 1
      Exit For
    End If
  Next

  If number_of_divisors = 2 Then
    Console.WriteLine("Number " & x & " is prime")
  End If
End Sub
```

14. Solution

```
Sub Main(args As String())
  Dim a, b, c, i, number_of_divisors, x As Integer

  Console.Write("Enter an integer greater than 1: ")
  a = Console.ReadLine()
  Do While a < 2
    Console.Write("Wrong number. Please enter an integer greater than 1: ")
    a = Console.ReadLine()
  Loop

  Console.Write("Enter a second integer greater than 1: ")
  b = Console.ReadLine()
  Do While b < 2
    Console.Write("Wrong number. Please enter a second integer greater than 1: ")
    b = Console.ReadLine()
  Loop

  If a > b Then
    c = a
    a = b
    b = c
  End If

  For x = a To b
    number_of_divisors = 2
    i = 2
```

```
    Do While i <= x \ 2 And number_of_divisors = 2
      If x Mod i = 0 Then
        number_of_divisors += 1
      End If
      i += 1
    Loop
    If number_of_divisors = 2 Then
      Console.WriteLine("Number " & x & " is prime")
    End If
  Next
End Sub
```

15. Solution

```
Sub Main(args As String())
  Dim a, b, c, d1, d2, d3, d4, r, x As Integer

  Console.Write("Enter a positive four-digit integer: ")
  a = Console.ReadLine()
  Do While a < 1000 Or a > 9999
    Console.Write("Wrong number. Please enter a positive four-digit integer: ")
    a = Console.ReadLine()
  Loop

  Console.Write("Enter a second positive four-digit integer: ")
  b = Console.ReadLine()
  Do While b < 1000 Or b > 9999
    Console.Write("Wrong number. Please enter a second positive four-digit integer: ")
    b = Console.ReadLine()
  Loop

  If a > b Then
    c = a
    a = b
    b = c
  End If

  For x = a To b
    d4 = x Mod 10
    r = x \ 10
    d3 = r Mod 10
    r = r \ 10
    d2 = r Mod 10
    d1 = r \ 10

    If d1 = d4 And d2 = d3 Then
      Console.WriteLine(x)
    End If
  Next
End Sub
```

16. Solution

```
Sub Main(args As String())
  Dim i As Integer

  For i = 0 To 30
    Console.WriteLine(2 ^ i)
  Next
End Sub
```

17. Solution

```
Sub Main(args As String())
  Dim i, offset As Integer

  offset = 10
  i = 1
  Do While i <= 401
    Console.WriteLine(i)
    i += offset
    offset += 2
  Loop
End Sub
```

18. Solution

```
Sub Main(args As String())
  Dim i As Integer

  For i = 1 To 100
    Console.WriteLine(-i & vbCrLf & i)
  Next
End Sub
```

19. Solution

First approach

```
Sub Main(args As String())
  Dim i, offset, value As Integer

  value = 0
  For i = 1 To 8
    offset = 10 ^ (i - 1)
    value += offset
    Console.WriteLine(value)
  Next
End Sub
```

Second approach

```
Sub Main(args As String())
  Dim i As Integer
  Dim value As String
```

```
  value = "1"
  For i = 1 To 8
    Console.WriteLine(value)
    value &= "1"
  Next
End Sub
```

20. Solution

```
Sub Main(args As String())
  Dim a, fib, fib_prev, fib_prev_prev, i As Integer

  a = Console.ReadLine()

  fib_prev_prev = 0
  fib_prev = 1
  fib = 1
  For i = 1 To a
    Console.WriteLine(fib)
    fib = fib_prev + fib_prev_prev
    fib_prev_prev = fib_prev
    fib_prev = fib
  Next
End Sub
```

21. Solution

```
Sub Main(args As String())
  Dim a, fib, fib_prev, fib_prev_prev As Integer

  a = Console.ReadLine()

  fib_prev_prev = 0
  fib_prev = 1
  fib = 1
  Do While fib < a
    Console.WriteLine(fib)
    fib = fib_prev + fib_prev_prev
    fib_prev_prev = fib_prev
    fib_prev = fib
  Loop
End Sub
```

22. Solution

```
Sub Main(args As String())
  Dim denominator, i, n, nominator As Integer
  Dim y As Double

  Console.Write("Enter a positive integer: ")
  n = Console.ReadLine()
  Do While n <= 0
    Console.Write("Wrong number. Please enter a positive integer: ")
```

```
    n = Console.ReadLine()
  Loop

  nominator = 0
  For i = 2 To 2 * n Step 2
    nominator += i
  Next

  denominator = 1
  For i = 1 To n
    denominator *= i
  Next

  y = nominator / denominator
  Console.WriteLine(y)
End Sub
```

23. Solution

```
Sub Main(args As String())
  Dim i, n, nominator, sign As Integer
  Dim y As Double

  Console.Write("Enter a positive integer: ")
  n = Console.ReadLine()
  Do While n <= 0
    Console.Write("Wrong number. Please enter a positive integer: ")
    n = Console.ReadLine()
  Loop

  nominator = 0
  sign = 1
  For i = 1 To 2 * n + 1 Step 2
    nominator += sign * i
    sign = -sign
  Next

  y = nominator / n
  Console.WriteLine(y)
End Sub
```

24. Solution

```
Sub Main(args As String())
  Dim i, n, sign As Integer
  Dim y As Double

  Console.Write("Enter a positive integer: ")
  n = Console.ReadLine()
  Do While n <= 0
    Console.Write("Wrong number. Please enter a positive integer: ")
    n = Console.ReadLine()
  Loop
```

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

  sign = 1
  For i = 3 To n Step 2
    y += sign / i
    sign = -sign
  Next

  Console.WriteLine(y)
End Sub
```

25. Solution

```
Sub Main(args As String())
  Dim i, n As Integer
  Dim y As Double

  Console.Write("Enter a positive integer: ")
  n = Console.ReadLine()
  Do While n <= 0
    Console.Write("Wrong number. Please enter a positive integer: ")
    n = Console.ReadLine()
  Loop

  y = 0
  For i = 1 To n
    y += 1 / i ^ (n - i + 1)
  Next

  Console.WriteLine(y)
End Sub
```

26. Solution

```
Sub Main(args As String())
  Dim factorial, i, n As Integer

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

  factorial = 1
  For i = 1 To n
    factorial *= i
  Next

  Console.WriteLine(factorial)
End Sub
```

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

27. Solution

First approach

```
Const ACCURACY = 0.00001

Sub Main(args As String())
  Dim i, j As Integer
  Dim factorial, exponential_previous, exponential, x As Double

  x = Console.ReadLine()

  exponential = 0
  i = 0
  Do
    exponential_previous = exponential

    factorial = 1
    For j = 1 To i
      factorial *= j
    Next

    exponential += x ^ i / factorial

    i += 1
  Loop While Math.Abs(exponential - exponential_previous) > ACCURACY

  Console.WriteLine("e(" & x & ") ~= " & exponential)
End Sub
```

Second approach

```
Const ACCURACY = 0.00001

Sub Main(args As String())
  Dim i As Integer
  Dim factorial, exponential_previous, exponential, x As Double

  x = Console.ReadLine()

  exponential = 1
  i = 1
  factorial = 1
  Do
    exponential_previous = exponential

    factorial *= i

    exponential += x ^ i / factorial

    i += 1
  Loop While Math.Abs(exponential - exponential_previous) > ACCURACY

  Console.WriteLine("e(" & x & ") ~= " & exponential)
End Sub
```

28. Solution

First approach

```
Const ACCURACY = 0.00001

Sub Main(args As String())
  Dim i, j, sign As Integer
  Dim factorial As Double
  Dim sinus, sinus_previous, x As Double

  x = Console.ReadLine()

  sign = 1
  sinus = 0
  i = 1
  Do
    sinus_previous = sinus

    factorial = 1
    For j = 1 To i
      factorial *= j
    Next

    sinus += sign * x ^ i / factorial

    sign = -sign
    i += 2
  Loop While Math.Abs(sinus - sinus_previous) > ACCURACY

  Console.WriteLine("sin(" & x & ") ~= " & sinus)
End Sub
```

Second approach

```
Const ACCURACY = 0.00001

Sub Main(args As String())
  Dim i, sign As Integer
  Dim factorial As Double
  Dim sinus, sinus_previous, x As Double

  x = Console.ReadLine()

  sign = -1
  sinus = x
  i = 3
  factorial = 1
  Do
    sinus_previous = sinus

    factorial *= i * (i - 1)

    sinus += sign * x ^ i / factorial

    sign = -sign
```

```
    i += 2
  Loop While Math.Abs(sinus - sinus_previous) > ACCURACY

  Console.WriteLine("sin(" & x & ") ~= " & sinus)
End Sub
```

29. Solution

First approach

```
Const ACCURACY = 0.00001

Sub Main(args As String())
  Dim i, j, sign As Integer
  Dim factorial As Double
  Dim cosinus, cosinus_previous, x As Double

  x = Console.ReadLine()

  sign = 1
  cosinus = 0
  i = 0
  Do
    cosinus_previous = cosinus

    factorial = 1
    For j = 1 To i
      factorial *= j
    Next

    cosinus += sign * x ^ i / factorial

    sign = -sign
    i += 2
  Loop While Math.Abs(cosinus - cosinus_previous) > ACCURACY

  Console.WriteLine("cos(" & x & ") ~= " & cosinus)
End Sub
```

Second approach

```
Const ACCURACY = 0.00001

Sub Main(args As String())
  Dim i, sign As Integer
  Dim factorial As Double
  Dim cosinus, cosinus_previous, x As Double

  x = Console.ReadLine()

  sign = -1
  cosinus = 1
  i = 2
  factorial = 1
  Do
    cosinus_previous = cosinus
```

```
    factorial *= i * (i - 1)

    cosinus += sign * x ^ i / factorial

    sign = -sign
    i += 2
  Loop While Math.Abs(cosinus - cosinus_previous) > ACCURACY

  Console.WriteLine("cos(" & x & ") ~= " & cosinus)
End Sub
```

30. Solution

```
Sub Main(args As String())
  Dim i As Integer
  Dim maximum, total, t As Double

  maximum = -460
  total = 0
  For i = 1 To 31
    Console.Write("Enter temperature for day " & i & ": ")
    t = Console.ReadLine()
    Do While t < -459.67
      Console.WriteLine("Error! Wrong temperature.")
      Console.Write("Enter temperature for day " & i & ": ")
      t = Console.ReadLine()
    Loop

    total += t
    If t > maximum Then
      maximum = t
    End If
  Next

  Console.WriteLine(total / 31 & " " & maximum)
End Sub
```

31. Solution

```
Sub Main(args As String())
  Dim hour, max_hour, max_minutes, min_hour, min_minutes, minutes As Integer
  Dim level, maximum, minimum As Double

  level = Console.ReadLine()
  If level <> 9999 Then
    hour = Console.ReadLine()
    minutes = Console.ReadLine()

    maximum = level
    max_hour = hour
    max_minutes = minutes

    minimum = level
    min_hour = hour
```

```
    min_minutes = minutes

    level = Console.ReadLine()
    Do While level <> 9999
      hour = Console.ReadLine()
      minutes = Console.ReadLine()

      If level > maximum Then
        maximum = level
        max_hour = hour
        max_minutes = minutes
      End If

      If level < minimum Then
        minimum = level
        min_hour = hour
        min_minutes = minutes
      End If

      level = Console.ReadLine()
    Loop

    Console.WriteLine(maximum & ", " & max_hour & ", " & max_minutes)
    Console.WriteLine(minimum & ", " & min_hour & ", " & min_minutes)
  End If
End Sub
```

32. Solution

```
Sub Main(args As String())
  Dim a, b, c, i As Integer
  Dim failure As Boolean

  Dim alphabet As String = "abcdefghijklmnopqrstuvwxyz"

  Do
    Console.Write("Enter an integer between 1 and 26: ")
    a = Console.ReadLine()

    failure = False
    If a < 1 Then
      Console.WriteLine("Please enter positive integers!")
      failure = True
    ElseIf a > 26 Then
      Console.WriteLine("Please enter a value less than or equal to 26!")
      failure = True
    End If
  Loop While failure

  Do
    Console.Write("Enter an integer between 1 and 26: ")
    b = Console.ReadLine()

    failure = False
```

```
    If b < 1 Then
      Console.WriteLine("Please enter positive integers!")
      failure = True
    ElseIf b > 26 Then
      Console.WriteLine("Please enter a value less than or equal to 26!")
      failure = True
    End If
  Loop While failure

  If a > b Then
    c = a
    a = b
    b = c
  End If

  For i = a To b
    Console.Write(alphabet(i - 1))
  Next
End Sub
```

33. Solution

```
Sub Main(args As String())
  Dim attempts, guess, secret_number As Integer

  Dim rnd As New Random()

  secret_number = rnd.Next(1, 101)

  attempts = 1
  Console.Write("Enter a guess: ")
  guess = Console.ReadLine()
  Do While guess <> secret_number
    If guess > secret_number Then
      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.")
    End If
    attempts += 1
    Console.Write("Enter a guess: ")
    guess = Console.ReadLine()
  Loop
  Console.WriteLine("You found it!")
  Console.WriteLine("Attempts: " & attempts)
End Sub
```

34. Solution

```
Sub Main(args As String())
  Dim attempts As Integer = 0
  Dim first_player_attempts, guess, i, secret_number As Integer

  Dim rnd As New Random()
```

```
  For i = 1 To 2
    secret_number = rnd.Next(1, 101)

    attempts = 1
    Console.Write("Enter a guess: ")
    guess = Console.ReadLine()
    Do While guess <> secret_number
      If guess > secret_number Then
        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.")
      End If
      attempts += 1
      Console.Write("Enter a guess: ")
      guess = Console.ReadLine()
    Loop
    Console.WriteLine("You found it!")
    Console.WriteLine("Attempts: " & attempts)

    If i = 1 Then
      first_player_attempts = attempts
    End If
  Next

  If first_player_attempts < attempts Then
    Console.WriteLine("First player wins!")
  ElseIf first_player_attempts > attempts Then
    Console.WriteLine("Second player wins!")
  Else
    Console.WriteLine("It's a draw")
  End If
End Sub
```

35. Solution

```
Sub Main(args As String())
  Dim choice, diagonal As Integer

  Do While True
    Console.WriteLine("1. 4/3 TV Screen")
    Console.WriteLine("2. 16/9 TV Screen")
    Console.WriteLine("3. Exit")
    Console.Write("Enter a choice: ")
    choice = Console.ReadLine()

    If choice = 3 Then
      Exit Do
    ElseIf choice = 1 Then
      Console.WriteLine("Enter diagonal: ")
      diagonal = Console.ReadLine()
      Console.WriteLine("Width: " & (diagonal * 0.8))
      Console.WriteLine("Height: " & (diagonal * 0.6))
```

```
    ElseIf choice = 2 Then
      Console.WriteLine("Enter diagonal: ")
      diagonal = Console.ReadLine()
      Console.WriteLine("Width: " & (diagonal * 0.87))
      Console.WriteLine("Height: " & (diagonal * 0.49))
    End If
  Loop
End Sub
```

36. Solution

```
Sub Main(args As String())
  Dim count_a, count_a_boys, count_b, count_cdef_girls, grade As Integer
  Dim i, maximum, minimum, n, total, total_a, total_a_boys, total_b As Integer
  Dim gender As String

  Console.Write("Enter total number of students: ")
  n = Console.ReadLine()
  Do While n <= 0
    Console.Write("Wrong number. Please enter total number of students: ")
    n = Console.ReadLine()
  Loop

  total = 0
  total_a = 0
  count_a = 0
  total_b = 0
  count_b = 0
  total_a_boys = 0
  count_a_boys = 0
  count_cdef_girls = 0

  maximum = -1
  minimum = 101

  For i = 1 To n
    Console.Write("Enter grade for student No " & i & ": ")
    grade = Console.ReadLine()
    Do While grade < 0 Or grade > 100
      Console.Write("Wrong grade. Please enter grade for student No " & i & ": ")
      grade = Console.ReadLine()
    Loop

    Console.Write("Enter gender for student No " & i & ": ")
    gender = Console.ReadLine().ToUpper()
    Do While gender <> "M" And gender <> "F"
      Console.Write("Wrong gender. Please enter gender for student No " & i & ": ")
      gender = Console.ReadLine().ToUpper()
    Loop

    If grade >= 90 And grade <= 100 Then
      total_a += grade
      count_a += 1
```

```
      If gender = "M" Then
        total_a_boys += grade
        count_a_boys += 1
      End If
    ElseIf grade >= 80 And grade <= 89 Then
      total_b += grade
      count_b += 1
    Else
      If gender = "F" Then
        count_cdef_girls += 1
      End If
    End If

    If grade > maximum Then
      maximum = grade
    End If

    If grade < minimum Then
      minimum = grade
    End If

    total += grade
  Next

  If count_a > 0 Then
    Console.Write("The average value of those who got an 'A' is: ")
    Console.WriteLine(total_a / count_a)
  End If
  If count_b > 0 Then
    Console.Write("The average value of those who got a 'B' is: ")
    Console.WriteLine(total_b / count_b)
  End If
  If count_a_boys > 0 Then
    Console.Write("The average value of boys who got an 'A' is: ")
    Console.WriteLine(total_a_boys / count_a_boys)
  End If
  Console.WriteLine("The total number of girls that got less than 'B' is: " & count_cdef_girls)
  Console.WriteLine("The highest grade is: " & maximum)
  Console.WriteLine("The lowest grade is: " & minimum)
  Console.WriteLine("The average grade of the whole class is: " & total / n)
End Sub
```

37. Solution

```
Sub Main(args As String())
  Dim amount, discount As Double
  Dim answer As String

  Do
    Console.Write("Enter amount: ")
    amount = Console.ReadLine()
    Do While amount <= 0
```

```
      Console.WriteLine("Wrong amount. Please enter amount: ")
      amount = Console.ReadLine()
    Loop

    If amount < 20 Then
      discount = 0
    ElseIf amount < 50 Then
      discount = 3
    ElseIf amount < 100 Then
      discount = 5
    Else
      discount = 10
    End If

    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()
  Loop While answer = "YES"
End Sub
```

38. Solution

```
Const TAX_RATE = 0.25

Sub Main(args As String())
  Dim kwh As Integer
  Dim t As Double

  Console.Write("Enter number of Kilowatt-hours consumed: ")
  kwh = Console.ReadLine()
  Do While kwh < 0 And kwh <> -1
    Console.Write("Wrong value. Please enter number of Kilowatt-hours consumed: ")
    kwh = Console.ReadLine()
  Loop

  Do While kwh <> -1
    If kwh <= 400 Then
      t = kwh * 0.11
    ElseIf kwh <= 1500 Then
      t = 400 * 0.11 + (kwh - 400) * 0.22
    ElseIf kwh <= 3500 Then
      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
    End If

    t += t * TAX_RATE
    Console.WriteLine("Total amount to pay (taxes included): " & t)

    Console.Write("Enter number of Kilowatt-hours consumed: ")
    kwh = Console.ReadLine()
```

```
    Do While kwh < 0 And kwh <> -1
      Console.Write("Wrong value. Please enter number of Kilowatt-hours consumed: ")
      kwh = Console.ReadLine()
    Loop
  Loop
End Sub
```

Review in “Loop Control Structures”

Review Crossword Puzzle

1.

Chapter 31

31.13 Review Questions: True/False

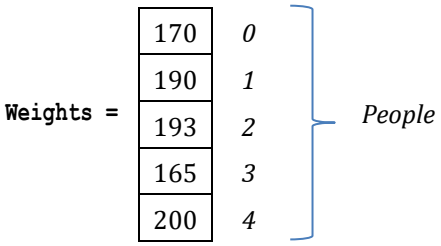
1. True
2. True
3. False
4. False
5. False
6. True
7. False
8. True
9. False
10. True
11. True
12. True
13. False
14. False
15. False
16. True
17. False
18. True
19. True
20. False
21. True
22. False
23. True
24. False
25. True
26. False
27. False
28. True
29. False
30. True
31. True
32. False
33. False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

31.14 Review Questions: Multiple Choice

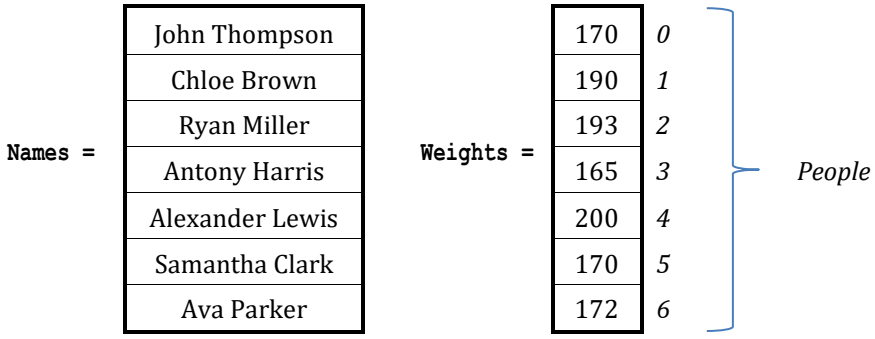
1. b
2. a
3. c
4. b
5. d
6. b
7. d
8. d
9. c
10. a
11. b
12. a
13. b
14. b

31.15 Review Exercises

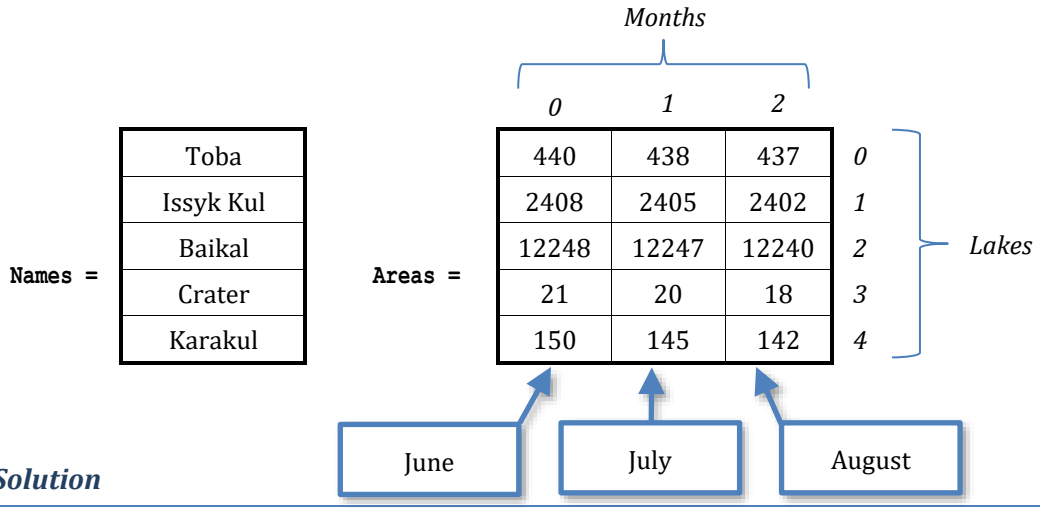
1. Solution



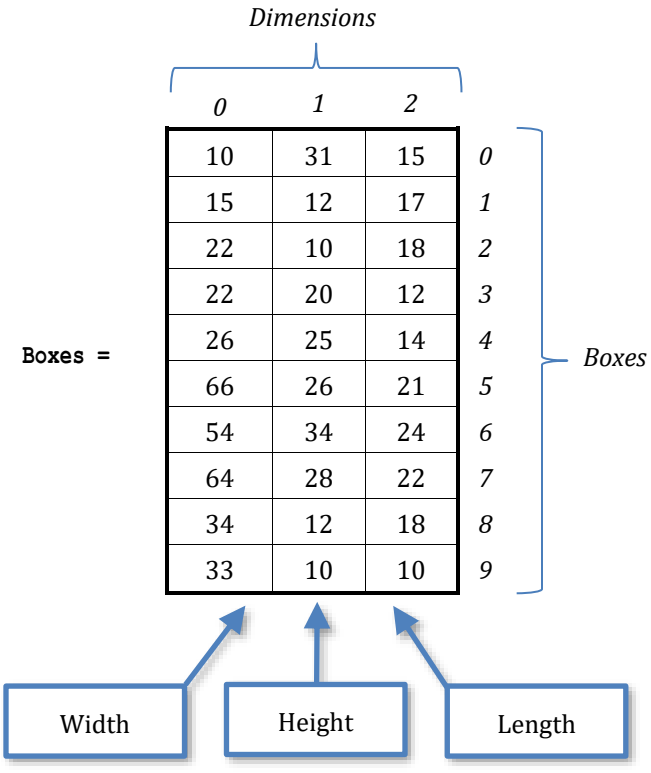
2. Solution



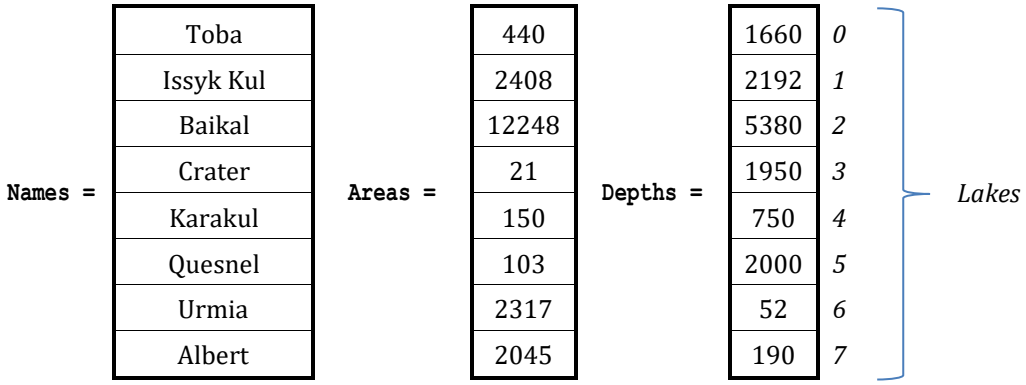
3. Solution



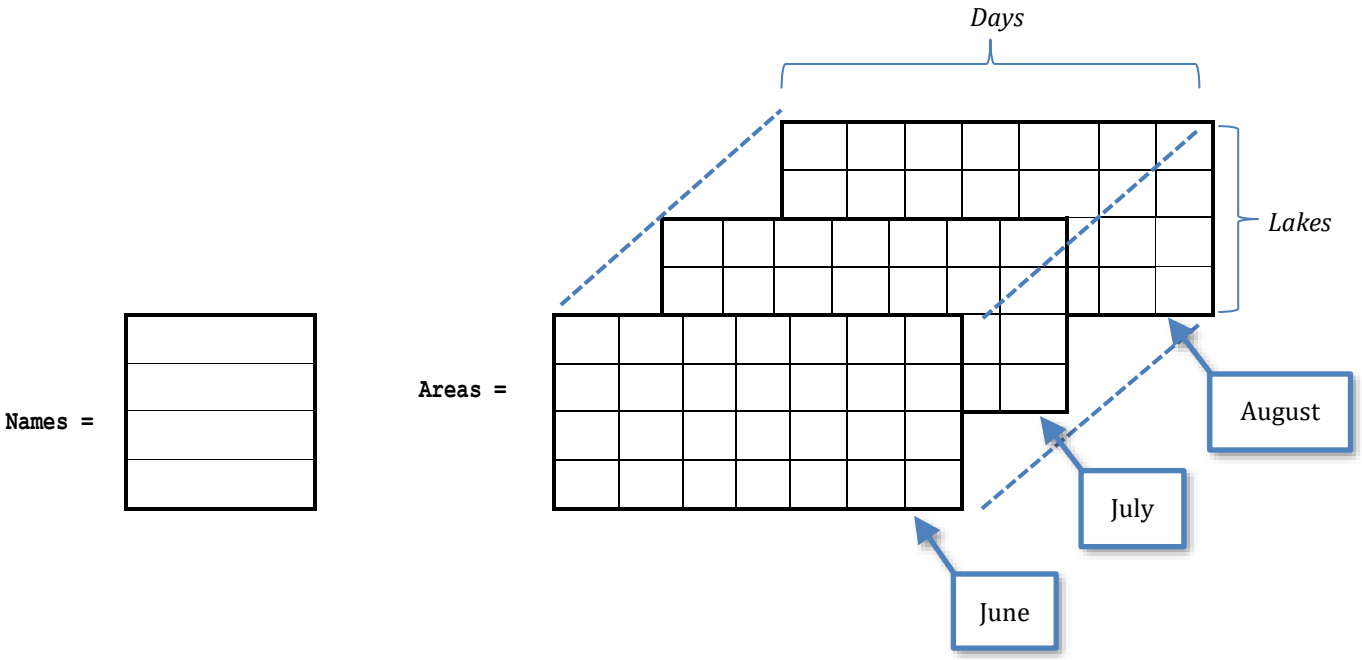
4. Solution



5. Solution



6. Solution



7. Solution

Step	Statement	x	a(0)	a(1)	a(2)
1	`Dim a(2) As Integer`	?	?	?	?
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	`Dim a(4) As Integer`	?	?	?	?	?	?
2	`a(1) = 5`	?	?	**5**	?	?	?
3	`x = 0`	**0**	?	5	?	?	?
4	`a(x) = 4`	0	**4**	5	?	?	?
5	`a(a(0)) = a(x + 1) Mod 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	`Dim a(3) As Integer`	?	?	?	?	?
2	`a(1) = Console.ReadLine()`	?	?	**3**	?	?
3	`x = 0`	**0**	?	3	?	?
4	`a(x) = 3`	0	**3**	3	?	?
5	`a(a(0)) = a(x + 1) Mod 2`	0	3	3	?	**1**
6	`a(a(0) Mod 2) = 10`	0	3	**10**	?	1
7	`x += 1`	**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	`Dim a(3) As Integer`	?	?	?	?	?
2	`a(1) = Console.ReadLine()`	?	?	**4**	?	?
3	`x = 0`	**0**	?	4	?	?
4	`a(x) = 3`	0	**3**	4	?	?
5	`a(a(0)) = a(x + 1) Mod 2`	0	3	4	?	**0**
6	`a(a(0) Mod 2) = 10`	0	3	**10**	?	0
7	`x += 1`	**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	`Dim a(3) As Integer`	?	?	?	?	?
2	`a(1) = Console.ReadLine()`	?	?	**1**	?	?
3	`x = 0`	**0**	?	1	?	?

4	a(x) = 3	0	**3**	1	?	?
5	a(a(0)) = a(x + 1) Mod 2	0	3	1	?	**3**
6	a(a(0) Mod 2) = 10	0	3	**10**	?	3
7	x += 1	**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	Dim a(3) As Integer	?	?	?	?	?
2	a(1) = Console.ReadLine()	?	?	**100**	?	?
3	x = 0	**0**	?	100	?	?
4	a(x) = 3	0	**3**	100	?	?
5	a(a(0)) = a(x + 1) Mod 10	0	3	100	?	**0**
6	If a(3) > 5 Then			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	Dim a(3) As Integer	?	?	?	?	?
2	a(1) = Console.ReadLine()	?	?	**108**	?	?
3	x = 0	**0**	?	108	?	?
4	a(x) = 3	0	**3**	108	?	?
5	a(a(0)) = a(x + 1) Mod 10	0	3	108	?	**8**
6	If a(3) > 5 Then			True		
7	a(a(0) Mod 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	Dim a(3) As Integer	?	?	?	?	?
2	a(1) = Console.ReadLine()	?	?	**1**	?	?
3	x = 0	**0**	?	1	?	?
4	a(x) = 3	0	**3**	1	?	?
5	a(a(0)) = a(x + 1) Mod 10	0	3	1	?	**1**
6	If a(3) > 5 Then			False		
7	a(2) = 3	0	3	1	**3**	1

11. Solution

Step	Statement	x	y	a(0)	a(1)	a(2)
1	`Dim a(2) As Integer`	?	?	?	?	?
2	`x = 4`	**4**	?	?	?	?
3	`y = x - 1`	4	**3**	?	?	?

4, 5	`If x > y Then`
	`  a(0) = 1`
	`Else`
	`  a(0) = y`
	`End If`
		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) Mod 2`	4	2	1	7	**1**

12. Solution

Step	Statement	i	a(0)	a(1)	a(2)	a(3)	a(4)	a(5)
1	`Dim a() As Integer = {17, 12, …`	?	**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 Then`	False						
5	`a(i) += 1`	0	**18**	12	45	12	12	49
6	`i += 1`	**1**	18	12	45	12	12	49
7	`i <= 5`	True						
8	`If a(i) = 12 Then`	True						
9	`a(i) -= 1`	1	18	**11**	45	12	12	49
10	`i += 1`	**2**	18	11	45	12	12	49
11	`i <= 5`	True						
12	`If a(i) = 12 Then`	False						
13	`a(i) += 1`	2	18	11	**46**	12	12	49
14	`i += 1`	**3**	18	11	46	12	12	49
15	`i <= 5`	True						
16	`If a(i) = 12 Then`	True						
17	`a(i) -= 1`	3	18	11	46	**11**	12	49
18	`i += 1`	**4**	18	11	46	11	12	49
19	`i <= 5`	True						
20	`If a(i) = 12 Then`	True						
21	`a(i) -= 1`	4	18	11	46	11	**11**	49
22	`i += 1`	**5**	18	11	46	11	11	49

23	`i <= 5`	True						
24	`If a(i) = 12 Then`	False						
25	`a(i) += 1`	5	18	11	46	11	11	**50**
26	`i += 1`	**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	`Dim a() As Integer = {10, 15, 12, …`	?	**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 += 1`	**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 += 1`	**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 += 1`	**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 += 1`	**5**	10	22	45	67	86	19
15	`i <= 4`	False						

14. Solution

It displays:
Navajo
Cherokee
Sioux

15. Solution

```
Const ELEMENTS = 100

Sub Main(args As String())
  Dim i As Integer

  Dim a(ELEMENTS - 1) As Double
  For i = 0 To ELEMENTS - 1
    a(i) = Console.ReadLine()
  Next

  For i = 0 To ELEMENTS - 1
```

```
    Console.WriteLine(a(i) ^ 3)
  Next
End Sub
```

16. Solution

```
Const ELEMENTS = 80

Sub Main(args As String())
  Dim i As Integer

  Dim a(ELEMENTS - 1) As Double
  For i = 0 To ELEMENTS - 1
    a(i) = Console.ReadLine()
  Next

  For i = 0 To ELEMENTS - 1
    a(i) = a(i) ^ 2
  Next

  For i = ELEMENTS - 1 To 0 Step -1
    Console.WriteLine(a(i))
  Next
End Sub
```

17. Solution

```
Const ELEMENTS = 90

Sub Main(args As String())
  Dim i As Integer

  Dim a(ELEMENTS - 1) As Integer
  For i = 0 To ELEMENTS - 1
    a(i) = Console.ReadLine()
  Next

  For i = ELEMENTS - 1 To 0 Step -1
    If a(i) Mod 5 = 0 Then
      Console.WriteLine(a(i))
    End If
  Next
End Sub
```

18. Solution

```
Const ELEMENTS = 50

Sub Main(args As String())
  Dim i As Integer

  Dim a(ELEMENTS - 1) As Integer
  For i = 0 To ELEMENTS - 1
    a(i) = Console.ReadLine()
  Next
```

```
  For i = 0 To ELEMENTS - 1
    If a(i) Mod 2 = 0 Or a(i) > 10 Then
      Console.WriteLine(a(i))
    End If
  Next
End Sub
```

19. Solution

```
Const ELEMENTS = 30

Sub Main(args As String())
  Dim i As Integer
  Dim total As Double

  Dim a(ELEMENTS - 1) As Double
  For i = 0 To ELEMENTS - 1
    a(i) = Console.ReadLine()
  Next

  total = 0
  For i = 0 To ELEMENTS - 1
    If a(i) > 0 Then
      total += a(i)
    End If
  Next
  Console.WriteLine(total)
End Sub
```

20. Solution

```
Const ELEMENTS = 50

Sub Main(args As String())
  Dim i, total As Integer

  Dim a(ELEMENTS - 1) As Integer
  For i = 0 To ELEMENTS - 1
    a(i) = Console.ReadLine()
  Next

  total = 0
  For i = 0 To ELEMENTS - 1
    If a(i) >= 10 And a(i) <= 99 Then
      total += a(i)
    End If
  Next
  Console.WriteLine(total)
End Sub
```

21. Solution

```
Const ELEMENTS = 40
```

```
Sub Main(args As String())
  Dim i As Integer
  Dim sum_neg, sum_pos As Double

  Dim a(ELEMENTS - 1) As Double
  For i = 0 To ELEMENTS - 1
    a(i) = Console.ReadLine()
  Next

  sum_pos = 0
  sum_neg = 0
  For i = 0 To ELEMENTS - 1
    If a(i) > 0  Then
      sum_pos += a(i)
    ElseIf a(i) < 0 Then
      sum_neg += a(i)
    End If
  Next
  Console.WriteLine(sum_pos & ", " & sum_neg)
End Sub
```

22. Solution

```
Const ELEMENTS = 20

Sub Main(args As String())
  Dim i As Integer
  Dim total As Double

  Dim a(ELEMENTS - 1) As Double
  For i = 0 To ELEMENTS - 1
    a(i) = Console.ReadLine()
  Next

  total = 0
  For i = 0 To ELEMENTS - 1
    total += a(i)
  Next
  Console.WriteLine(total / ELEMENTS)
End Sub
```

23. Solution

```
Const ELEMENTS = 50

Sub Main(args As String())
  Dim i As Integer

  Dim a(ELEMENTS - 1) As Integer
  For i = 0 To ELEMENTS - 1
    Console.Write("Enter an integer: ")
    a(i) = Console.ReadLine()
  Next

  For i = 0 To ELEMENTS - 1
```

```
    If a(i) < 20 Then
      Console.WriteLine(a(i))
    End If
  Next
End Sub
```

24. Solution

```
Const ELEMENTS = 60

Sub Main(args As String())
  Dim i As Integer

  Dim a(ELEMENTS - 1) As Double
  For i = 0 To ELEMENTS - 1
    Console.Write("Enter a number: ")
    a(i) = Console.ReadLine()
  Next

  For i = 0 To ELEMENTS - 1 Step 2
    Console.WriteLine(a(i))
  Next
End Sub
```

25. Solution

```
Const ELEMENTS = 20

Sub Main(args As String())
  Dim i As Integer
  Dim total As Double

  Dim a(ELEMENTS - 1) As Double
  For i = 0 To ELEMENTS - 1
    Console.Write("Enter a number: ")
    a(i) = Console.ReadLine()
  Next

  total = 0
  For i = 0 To ELEMENTS - 1 Step 2
    total += a(i)
  Next
  Console.WriteLine(total)
End Sub
```

26. Solution

```
Const ELEMENTS = 100

Sub Main(args As String())
  Dim i As Integer
  Dim a(ELEMENTS - 1) As Integer
  For i = 0 To ELEMENTS - 1
    a(i) = i + 1
  Next
…
```

27. Solution

First approach

```
Const ELEMENTS = 100

Sub Main(args As String())
  Dim i, k As Integer
  Dim a(ELEMENTS - 1) As Integer
  k = 2
  For i = 0 To ELEMENTS - 1
    a(i) = k
    k += 2
  Next
…
```

Second approach

```
Const ELEMENTS = 100

Sub Main(args As String())
  Dim i As Integer
  Dim a(ELEMENTS - 1) As Integer
  For i = 0 To ELEMENTS - 1
    a(i) = (i + 1) * 2
  Next
…
```

28. Solution

```
Sub Main(args As String())
  Dim i, n As Integer

  Console.Write("Enter N: ")
  n = Console.ReadLine()

  Dim a(n - 1) As Integer
  For i = 1 To n
    a(i - 1) = i ^ 2
  Next

  For i = 0 To n - 1
    Console.WriteLine(a(i))
  Next
End Sub
```

29. Solution

```
Const ELEMENTS = 10

Sub Main(args As String())
  Dim i As Integer

  Dim a(ELEMENTS - 1) As Double
  For i = 1 To ELEMENTS - 1
    Console.Write("Enter a number: ")
    a(i) = Console.ReadLine()
  Next
```

```
  For i = 0 To ELEMENTS - 1
    If a(i) = Fix(a(i)) Then
      Console.WriteLine(i)
    End If
  Next
End Sub
```

30. Solution

```
Const ELEMENTS = 50

Sub Main(args As String())
  Dim i, count As Integer

  Dim a(ELEMENTS - 1) As Double
  For i = 1 To ELEMENTS - 1
    Console.Write("Enter a number: ")
    a(i) = Console.ReadLine()
  Next

  count = 0
  For i = 0 To ELEMENTS - 1
    If a(i) < 0 Then
      count += 1
    End If
  Next
  Console.WriteLine(count)
End Sub
```

31. Solution

```
Const WORDS = 50

Sub Main(args As String())
  Dim i As Integer

  Dim a(WORDS - 1) As String
  For i = 0 To WORDS - 1
    a(i) = Console.ReadLine()
  Next

  For i = 0 To WORDS - 1
    If a(i).Length >= 10  Then
      Console.WriteLine(a(i))
    End If
  Next
End Sub
```

32. Solution

```
Const ELEMENTS = 30

Sub Main(args As String())
  Dim i, k As Integer
```

```
  Dim words(ELEMENTS - 1) As String
  For i = 0 To ELEMENTS - 1
    words(i) = Console.ReadLine()
  Next

  Dim length_limits() As Integer = {0, 5, 10, 20}

  For k = 1 To 3
    For i = 0 To ELEMENTS - 1
      If words(i).Length >= length_limits(k - 1) And words(i).Length < length_limits(k) Then
        Console.WriteLine(words(i))
      End If
    Next
  Next
End Sub
```

33. Solution

```
Const WORDS = 40

Sub Main(args As String())
  Dim count, i, j As Integer

  Dim a(WORDS - 1) As String
  For i = 0 To WORDS - 1
    Console.Write("Enter a word: ")
    a(i) = Console.ReadLine()
  Next

  For i = 0 To WORDS - 1
    count = 0
    For j = 0 To a(i).Length - 1
      If a(i).Substring(j, 1) = "w" Then  'Alternatively use: If a(i)(j) = "w" Then
        count += 1
      End If
    Next
    If count >= 2 Then
      Console.WriteLine(a(i))
    End If
  Next
End Sub
```

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	`Dim a(1, 2) As Integer`	?	? ? ? ? ? ?
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	`Dim a(1, 2) As Integer`	?	?	? ? ? ? ? ?
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 += 1`	0	**1**	5 ? ? ? ? ?
8	`j <= 2`			True
9	`a(i, j) = (i + 1) * 5 + j`	0	1	5 **6** ? ? ? ?
10	`j += 1`	0	**2**	5 6 ? ? ? ?
11	`j <= 2`			True
12	`a(i, j) = (i + 1) * 5 + j`	0	2	5 6 **7** ? ? ?
13	`j += 1`	0	**3**	5 6 7 ? ? ?
14	`j <= 2`			False
15	`i += 1`	**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	**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`	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`	1	**3**	5 6 7 10 11 12
27	`j <= 2`	False		
28	`i += 1`	**2**	3	5 6 7 10 11 12
29	`i <= 1`	False		

3. Solution

Step	Statement	i	j	a
1	`Dim a(2, 2) As Integer`	?	?	? ? ? ? ? ? ? ? ?
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`	**1**	0	2 ? ? ? ? ? ? ? ?
8	`i <= 2`	True		
9	`a(i, j) = (i + 1) * 2 + j * 4`	1	0	2 ? ? **4** ? ? ? ? ?

10	`i += 1`	**2**	0	2 ? ? 4 ? ? ? ? ?
11	`i <= 2`			True
12	`a(i, j) = (i + 1) * 2 + j * 4`	2	0	2 ? ? 4 ? ? **6** ? ?
13	`i += 1`	**3**	0	2 ? ? 4 ? ? 6 ? ?
14	`i <= 2`			False
15	`j += 1`	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**	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 += 1`	**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 += 1`	**3**	1	2 6 ? 4 8 ? 6 10 ?
27	`i <= 2`			False
28	`j += 1`	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`	**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 += 1`	**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 += 1`	**3**	2	2 6 10 4 8 12 6 10 14
40	`i <= 2`			False
41	`j += 1`	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 ROWS = 10
Const COLUMNS = 15

Sub Main(args As String())
  Dim i, j As Integer

  Dim a(ROWS - 1, COLUMNS - 1) As Integer
  For i = 0 To ROWS - 1
    For j = 0 To COLUMNS - 1
      a(i, j) = Console.ReadLine()
    Next
  Next

  For i = 0 To ROWS - 1
    For j = 0 To COLUMNS - 1
      If a(i, j) Mod 2 <> 0 Then
        Console.WriteLine(i & ", " & j)
      End If
    Next
  Next
End Sub
```

10. Solution

```
Const ROWS = 10
Const COLUMNS = 6

Sub Main(args As String())
  Dim i, j As Integer

  Dim a(ROWS - 1, COLUMNS - 1) As Double
  For i = 0 To ROWS - 1
    For j = 0 To COLUMNS - 1
      a(i, j) = Console.ReadLine()
    Next
  Next

  For i = 0 To ROWS - 1
    For j = 0 To COLUMNS - 1 Step 2
      Console.WriteLine(a(i, j))
    Next
  Next
End Sub
```

11. Solution

```
Const ROWS = 12
Const COLUMNS = 8

Sub Main(args As String())
  Dim i, j As Integer
  Dim total As Double

  Dim a(ROWS - 1, COLUMNS - 1) As Double
  For i = 0 To ROWS - 1
    For j = 0 To COLUMNS - 1
      a(i, j) = Console.ReadLine()
    Next
  Next

  total = 0
  For i = 1 To ROWS - 1 Step 2
    For j = 0 To COLUMNS - 1 Step 2
      total += a(i, j)
    Next
  Next
  Console.WriteLine(total)
End Sub
```

12. Solution

```
Const N = 8

Sub Main(args As String())
  Dim i, j, k As Integer
  Dim sum_antidiagonal, sum_diagonal As Double

  Dim a(N - 1, N - 1) As Double
  For i = 0 To N - 1
    For j = 0 To N - 1
      a(i, j) = Console.ReadLine()
    Next
  Next

  sum_diagonal = 0
  sum_antidiagonal = 0
  For k = 0 To N - 1
    sum_diagonal += a(k, k)
    sum_antidiagonal += a(k, N - k - 1)
  Next
  Console.WriteLine(sum_diagonal / N & ", " & sum_antidiagonal / N)
End Sub
```

13. Solution

```
Const N = 5
```

```
Sub Main(args As String())
  Dim i, j As Integer

  Dim a(N - 1, N - 1) As Integer
  For i = 0 To N - 1
    For j = 0 To N - 1
      If i = N - j - 1 Then
        a(i, j) = 5
      ElseIf i > N - j - 1 Then
        a(i, j) = 88
      Else
        a(i, j) = 11
      End If
    Next
  Next

  For i = 0 To N - 1
    For j = 0 To N - 1
      Console.Write(a(i, j) & vbTab)
    Next
    Console.WriteLine()
  Next
End Sub
```

14. Solution

```
Const N = 5

Sub Main(args As String())
  Dim i, j As Integer

  Dim a(N - 1, N - 1) As Integer
  For i = 0 To N - 1
    For j = 0 To N - 1
      If i = N - j - 1 Then
        a(i, j) = 5
      ElseIf i > N - j - 1 Then
        a(i, j) = 88
      Else
        a(i, j) = 11
      End If
      If i = j Then
        a(i, j) = 0
      End If
    Next
  Next

  For i = 0 To N - 1
    For j = 0 To N - 1
      Console.Write(a(i, j) & vbTab)
    Next
    Console.WriteLine()
```

```
  Next
End Sub
```

15. Solution

```
Const ROWS = 5
Const COLUMNS = 4

Sub Main(args As String())
  Dim i, j As Integer

  Dim a(ROWS - 1, COLUMNS - 1) As Double
  For i = 0 To ROWS - 1
    For j = 0 To COLUMNS - 1
      a(i, j) = Console.ReadLine()
    Next
  Next

  For i = 0 To ROWS - 1
    For j = 0 To COLUMNS - 1
      If a(i, j) = Fix(a(i, j)) Then
        Console.WriteLine(i & ", " & j)
      End If
    Next
  Next
End Sub
```

16. Solution

```
Const ROWS = 10
Const COLUMNS = 4

Sub Main(args As String())
  Dim count, i, j As Integer

  Dim a(ROWS - 1, COLUMNS - 1) As Double
  For i = 0 To ROWS - 1
    For j = 0 To COLUMNS - 1
      a(i, j) = Console.ReadLine()
    Next
  Next

  count = 0
  For i = 0 To ROWS - 1
    For j = 0 To COLUMNS - 1
      If a(i, j) < 0 Then
        count += 1
      End If
    Next
  Next
  Console.WriteLine(count)
End Sub
```

17. Solution

```
Const ROWS = 3
Const COLUMNS = 4

Sub Main(args As String())
  Dim i, j As Integer

  Dim a(ROWS - 1, COLUMNS - 1) As String
  For i = 0 To ROWS - 1
    For j = 0 To COLUMNS - 1
      a(i, j) = Console.ReadLine()
    Next
  Next

  For i = 0 To ROWS - 1
    For j = 0 To COLUMNS - 1
      Console.Write(a(i, j) & " ")
    Next
  Next
End Sub
```

18. Solution

```
Const ROWS = 20
Const COLUMNS = 14

Sub Main(args As String())
  Dim i, j As Integer

  Dim a(ROWS - 1, COLUMNS - 1) As String
  For i = 0 To ROWS - 1
    For j = 0 To COLUMNS - 1
      a(i, j) = Console.ReadLine()
    Next
  Next

  For i = 0 To ROWS - 1
    For j = 0 To COLUMNS - 1
      If a(i, j).Length < 5 Then
        Console.WriteLine(a(i, j))
      End If
    Next
  Next
End Sub
```

19. Solution

First approach

```
Const ROWS = 20
Const COLUMNS = 14

Sub Main(args As String())
```

```
  Dim i, j, k As Integer

  Dim a(ROWS - 1, COLUMNS - 1) As String
  For i = 0 To ROWS - 1
    For j = 0 To COLUMNS - 1
      a(i, j) = Console.ReadLine()
    Next
  Next

  Dim length_limits() As Integer = {5, 10, 20}

  For k = 0 To 2
    For i = 0 To ROWS - 1
      For j = 0 To COLUMNS - 1
        If a(i, j).Length < length_limits(k) Then
          Console.WriteLine(a(i, j))
        End If
      Next
    Next
  Next
End Sub
```

Second approach

```
Const ROWS = 20
Const COLUMNS = 14

Sub Main(args As String())
  Dim i, j, k As Integer

  Dim a As New string(ROWS,  COLUMNS)
  For i = 0 To ROWS - 1
    For j = 0 To COLUMNS - 1
      a(i, j) = Console.ReadLine()
    Next
  Next

  For k = 0 To 2
    For i = 0 To ROWS - 1
      For j = 0 To COLUMNS - 1
        If a(i, j).Length < 5 * 2 ^ k Then
          Console.WriteLine(a(i, j))
        End If
      Next
    Next
  Next
End Sub
```

Chapter 33

33.7 Review Questions: True/False

1. True
2. False
3. True
4. False
5. False
6. False
7. True
8. True
9. False
10. False
11. True

33.8 Review Questions: Multiple Choice

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

33.9 Review Exercises

1. Solution

```
Const STUDENTS = 15
Const TESTS = 5

Sub Main(args As String())
  Dim i, j As Integer

  Dim grades(STUDENTS - 1, TESTS - 1) As Integer
  For i = 0 To STUDENTS - 1
    For j = 0 To TESTS - 1
      grades(i, j) = Console.ReadLine()
    Next
  Next

  Dim average(STUDENTS - 1) As Double
  For i = 0 To STUDENTS - 1
    average(i) = 0
    For j = 0 To TESTS - 1
      average(i) += grades(i, j)
    Next
    average(i) /= TESTS
  Next

  For i = 0 To STUDENTS - 1
    Console.Write("Student No " & (i + 1) & ": ")

    If average(i) < 60 Then
      Console.WriteLine("E/F")
    ElseIf average(i) < 70 Then
      Console.WriteLine("D")
    ElseIf average(i) < 80 Then
      Console.WriteLine("C")
    ElseIf average(i) < 90 Then
```

```
      Console.WriteLine("B")
    Else
      Console.WriteLine("A")
    End If
  Next
End Sub
```

2. Solution

```
Const OBJECTS = 5
Const FALLS = 10

Sub Main(args As String())
  Dim i, j, total As Integer

  Dim g(OBJECTS - 1, FALLS - 1) As Integer
  For i = 0 To OBJECTS - 1
    For j = 0 To FALLS - 1
      g(i, j) = Console.ReadLine()
    Next
  Next

  For i = 0 To OBJECTS - 1
    total = 0
    For j = 0 To FALLS - 1
      total += g(i, j)
    Next
    Console.WriteLine("Average g for object No " & (i + 1) & ": " & (total / FALLS))
  Next

  For j = 0 To FALLS - 1
    total = 0
    For i = 0 To OBJECTS - 1
      total += g(i, j)
    Next
    Console.WriteLine("Average g for fall No " & (j + 1) & ": " & (total / OBJECTS))
  Next

  total = 0
  For i = 0 To OBJECTS - 1
    For j = 0 To FALLS - 1
      total += g(i, j)
    Next
  Next
  Console.WriteLine("Overall average g: " & (total / (OBJECTS * FALLS)))
End Sub
```

3. Solution

```
Const PLAYERS = 15
Const MATCHES = 12

Sub Main(args As String())
```

```
  Dim i, j, total As Integer

  Dim points(PLAYERS - 1, MATCHES - 1) As Integer
  For i = 0 To PLAYERS - 1
    For j = 0 To MATCHES - 1
      points(i, j) = Console.ReadLine()
    Next
  Next

  For i = 0 To PLAYERS - 1
    total = 0
    For j = 0 To MATCHES - 1
      total += points(i, j)
    Next
    Console.WriteLine("Total number of points for player No " & (i + 1) & ": " & total)
  Next

  For j = 0 To MATCHES - 1
    total = 0
    For i = 0 To PLAYERS - 1
      total += points(i, j)
    Next
    Console.WriteLine("Total number of points for match No " & (j + 1) & ": " & total)
  Next
End Sub
```

4. Solution

```
Const CITIES = 20
Const HOURS = 24

Sub Main(args As String())
  Dim i, j As Integer
  Dim total As Double

  Dim temperatures(CITIES - 1, HOURS - 1) As Double
  For i = 0 To CITIES - 1
    For j = 0 To HOURS - 1
      temperatures(i, j) = Console.ReadLine()
    Next
  Next

  For j = 0 To HOURS - 1
    total = 0
    For i = 0 To CITIES - 1
      total += temperatures(i, j)
    Next
    If total / CITIES < 10 Then
      Console.WriteLine("Hour: " & (j + 1))
    End If
  Next
End Sub
```

5. Solution

```
Const PLAYERS = 24
Const MATCHES = 10

Sub Main(args As String())
  Dim i, j, total As Integer

  Dim names(PLAYERS - 1) As String
  Dim goals(PLAYERS - 1, MATCHES - 1) As Integer
  For i = 0 To PLAYERS - 1
    names(i) = Console.ReadLine()
    For j = 0 To MATCHES - 1
      goals(i, j) = Console.ReadLine()
    Next
  Next

  For i = 0 To PLAYERS - 1
    total = 0
    For j = 0 To MATCHES - 1
      total += goals(i, j)
    Next
    Console.WriteLine(names(i) & ": " & (total / MATCHES))
  Next

  For j = 0 To MATCHES - 1
    total = 0
    For i = 0 To PLAYERS - 1
      total += goals(i, j)
    Next
    Console.WriteLine("Match No " & (j + 1) & ": " & total)
  Next
End Sub
```

6. Solution

```
Const STUDENTS = 12
Const LESSONS = 6

Sub Main(args As String())
  Dim i, j, total As Integer

  Dim names(STUDENTS - 1) As String
  Dim grades(STUDENTS - 1, LESSONS - 1) As Integer
  For i = 0 To STUDENTS - 1
    names(i) = Console.ReadLine()
    For j = 0 To LESSONS - 1
      grades(i, j) = Console.ReadLine()
    Next
  Next

  Dim average(STUDENTS - 1) As Double
  For i = 0 To STUDENTS - 1
```

```
    total = 0
    For j = 0 To LESSONS - 1
      total += grades(i, j)
    Next
    average(i) = total / LESSONS
    Console.WriteLine(names(i) & ": " & average(i))
  Next

  For j = 0 To LESSONS - 1
    total = 0
    For i = 0 To STUDENTS - 1
      total += grades(i, j)
    Next
    Console.WriteLine(total / STUDENTS)
  Next

  For i = 0 To STUDENTS - 1
    If average(i) < 60 Then
      Console.WriteLine(names(i))
    End If
  Next

  For i = 0 To STUDENTS - 1
    If average(i) > 89 Then
      Console.WriteLine(names(i) & " Bravo!")
    End If
  Next
End Sub
```

7. Solution

```
Const ARTISTS = 15
Const JUDGES = 5

Sub Main(args As String())
  Dim i, j, total As Integer

  Dim judge_names(JUDGES - 1) As String
  For j = 0 To JUDGES - 1
    Console.Write("Enter name for judge No " & (j + 1) & ": ")
    judge_names(j) = Console.ReadLine()
  Next

  Dim artist_names(ARTISTS - 1) As String
  Dim song_titles(ARTISTS - 1) As String
  Dim score(ARTISTS - 1, JUDGES - 1) As Integer
  For i = 0 To ARTISTS - 1
    Console.Write("Enter name for artist No " & (i + 1) & ": ")
    artist_names(i) = Console.ReadLine()
    Console.Write("Enter song title for artist " & artist_names(i) & ": ")
    song_titles(i) = Console.ReadLine()
    For j = 0 To JUDGES - 1
      Console.Write("Enter score for artist: " & artist_names(i))
```

```
      Console.Write(" gotten from judge " & judge_names(j) & ": ")
      score(i, j) = Console.ReadLine()
    Next
  Next

  For i = 0 To ARTISTS - 1
    total = 0
    For j = 0 To JUDGES - 1
      total += score(i, j)
    Next
    Console.WriteLine(artist_names(i) & ", " & song_titles(i) & ": " & total)
  Next

  For j = 0 To JUDGES - 1
    total = 0
    For i = 0 To ARTISTS - 1
      total += score(i, j)
    Next
    Console.WriteLine(judge_names(j) & ": " & total / ARTISTS)
  Next
End Sub
```

8. Solution

```
Const PEOPLE = 30
Const MONTHS = 12

Sub Main(args As String())
  Dim i, j, sum_heights, sum_weights As Integer
  Dim average_height, average_weight As Double

  Dim weights(PEOPLE - 1, MONTHS - 1) As Integer
  Dim heights(PEOPLE - 1, MONTHS - 1) As Integer
  For i = 0 To PEOPLE - 1
    For j = 0 To MONTHS - 1
      weights(i, j) = Console.ReadLine()
      heights(i, j) = Console.ReadLine()
    Next
  Next

  For i = 0 To PEOPLE - 1
    sum_weights = 0
    sum_heights = 0
    For j = 0 To MONTHS - 1
      sum_weights += weights(i, j)
      sum_heights += heights(i, j)
    Next
    average_weight = sum_weights / MONTHS
    average_height = sum_heights / MONTHS
    Console.WriteLine(average_weight & ", " & average_height)
    Console.WriteLine(average_weight * 702 / average_height ^ 2)
  Next
```

```
  For i = 0 To PEOPLE - 1
    Console.WriteLine(weights(i, 4) * 702 / heights(i, 4) ^ 2)
    Console.WriteLine(weights(i, 7) * 702 / heights(i, 7) ^ 2)
  Next
End Sub
```

9. Solution

```
Const VAT = 0.19
Const CONSUMERS = 1000

Sub Main(args As String())
  Dim consumed, i As Integer
  Dim payment, total As Double

  Dim meter_reading(CONSUMERS - 1, 1) As Integer
  For i = 0 To CONSUMERS - 1
    meter_reading(i, 0) = Console.ReadLine()
    meter_reading(i, 1) = Console.ReadLine()
  Next

  total = 0
  For i = 0 To CONSUMERS - 1
    consumed = meter_reading(i, 1) - meter_reading(i, 0)
    Console.WriteLine(consumed)
    payment = consumed * 0.07
    payment += VAT * payment
    Console.WriteLine(payment)

    total += consumed
  Next

  Console.WriteLine(total & ", " & (total * 0.07 + total * 0.07 * VAT))
End Sub
```

10. Solution

```
Const CURRENCIES = 4
Const DAYS = 5

Sub Main(args As String())
  Dim i, j As Integer
  Dim average, total, usd As Double

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

  Dim currency() As String = {
    "British Pounds Sterling", "Euros", "Canadian Dollars", "Australian Dollars"
  }

  Dim rate(,) As Double = {
    {1.320, 1.321, 1.332, 1.331, 1.341},
    {1.143, 1.156, 1.138, 1.122, 1.129},
```

```
    {0.757, 0.764, 0.760, 0.750, 0.749},
    {0.720, 0.725, 0.729, 0.736, 0.739}
  }

  For i = 0 To CURRENCIES - 1
    total = 0
    For j = 0 To DAYS - 1
      total += rate(i, j)
    Next
    average = total / DAYS
    Console.WriteLine(usd & " US dollars = " & (usd / average) & " " & currency(i))
  Next
End Sub
```

11. Solution

```
Const EMPLOYEES = 10
Const DAYS = 5

Sub Main(args As String())
  Dim i, j As Integer
  Dim gross_pay, pay_rate, total As Double

  Dim days() As String = {"Monday", "Tuesday", "Wednesday", "Thursday", "Friday"}

  pay_rate = Console.ReadLine()

  Dim names(EMPLOYEES - 1) As String
  Dim hours_worked_per_day(EMPLOYEES - 1, DAYS - 1) As Integer
  For i = 0 To EMPLOYEES - 1
    names(i) = Console.ReadLine()
    For j = 0 To DAYS - 1
      hours_worked_per_day(i, j) = Console.ReadLine()
    Next
  Next

  Dim hours_worked_per_week(EMPLOYEES - 1) As Integer
  For i = 0 To EMPLOYEES - 1
    hours_worked_per_week(i) = 0
    For j = 0 To DAYS - 1
      hours_worked_per_week(i) += hours_worked_per_day(i, j)
    Next
    If hours_worked_per_week(i) > 40 Then
      Console.WriteLine(names(i))
    End If
  Next

  For i = 0 To EMPLOYEES - 1
    If hours_worked_per_week(i) <= 40 Then
      gross_pay = pay_rate * hours_worked_per_week(i)
    Else
      gross_pay = pay_rate * 40 + 1.5 * pay_rate * (hours_worked_per_week(i) - 40)
    End If
```

```
    Console.WriteLine(names(i) & ", " & gross_pay)
  Next

  For i = 0 To EMPLOYEES - 1
    If hours_worked_per_week(i) > 40 Then
      For j = 0 To DAYS - 1
        If hours_worked_per_day(i, j) > 8 Then
          Console.WriteLine(names(i) & ", " & days(j) & " Overtime!")
        End If
      Next
    End If
  Next

  For j = 0 To DAYS - 1
    total = 0
    For i = 0 To EMPLOYEES - 1
      If hours_worked_per_day(i, j) <= 8 Then
        gross_pay = pay_rate * hours_worked_per_day(i, j)
      Else
        gross_pay = pay_rate * 8 + 1.5 * pay_rate * (hours_worked_per_day(i, j) - 8)
      End If
      total += gross_pay
    Next
    Console.WriteLine(days(j) & ", " & total)
  Next
End Sub
```

12. Solution

```
Const ROWS = 3
Const COLUMNS = 4

Sub Main(args As String())
  Dim i, j, k As Integer

  Dim a(,) As Integer = {
    {9, 9, 2, 6},
    {4, 1, 10, 11},
    {12, 15, 7, 3}
  }

  Dim b(ROWS * COLUMNS - 1) As Integer
  k = 0
  For i = 0 To ROWS - 1
    For j = 0 To COLUMNS - 1
      b(k) = a(i, j)
      k += 1
    Next
  Next

  For k = 0 To b.length - 1
    Console.Write(b(k) & " ")
  Next
```

```
End Sub
```

13. Solution

```
Const ROWS = 3
Const COLUMNS = 3

Sub Main(args As String())
  Dim i, j, k As Integer

  Dim a() As Integer = {16, 12, 3, 5, 6, 9, 18, 19, 20}

  Dim b(ROWS - 1, COLUMNS - 1) As Integer
  k = 0
  For i = ROWS - 1 To 0 Step -1
    For j = 0 To COLUMNS - 1
      b(i, j) = a(k)
      k += 1
    Next
  Next

  For i = 0 To ROWS - 1
    For j = 0 To COLUMNS - 1
      Console.Write(b(i, j) & vbTab)
    Next
    Console.WriteLine()
  Next
End Sub
```

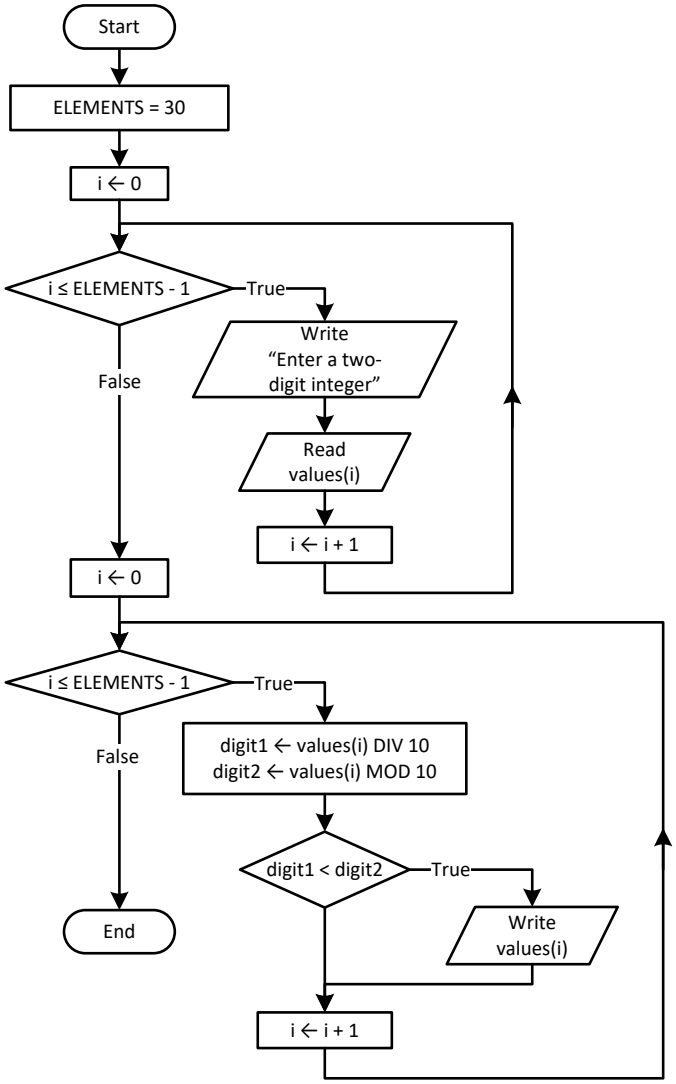
Chapter 34

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

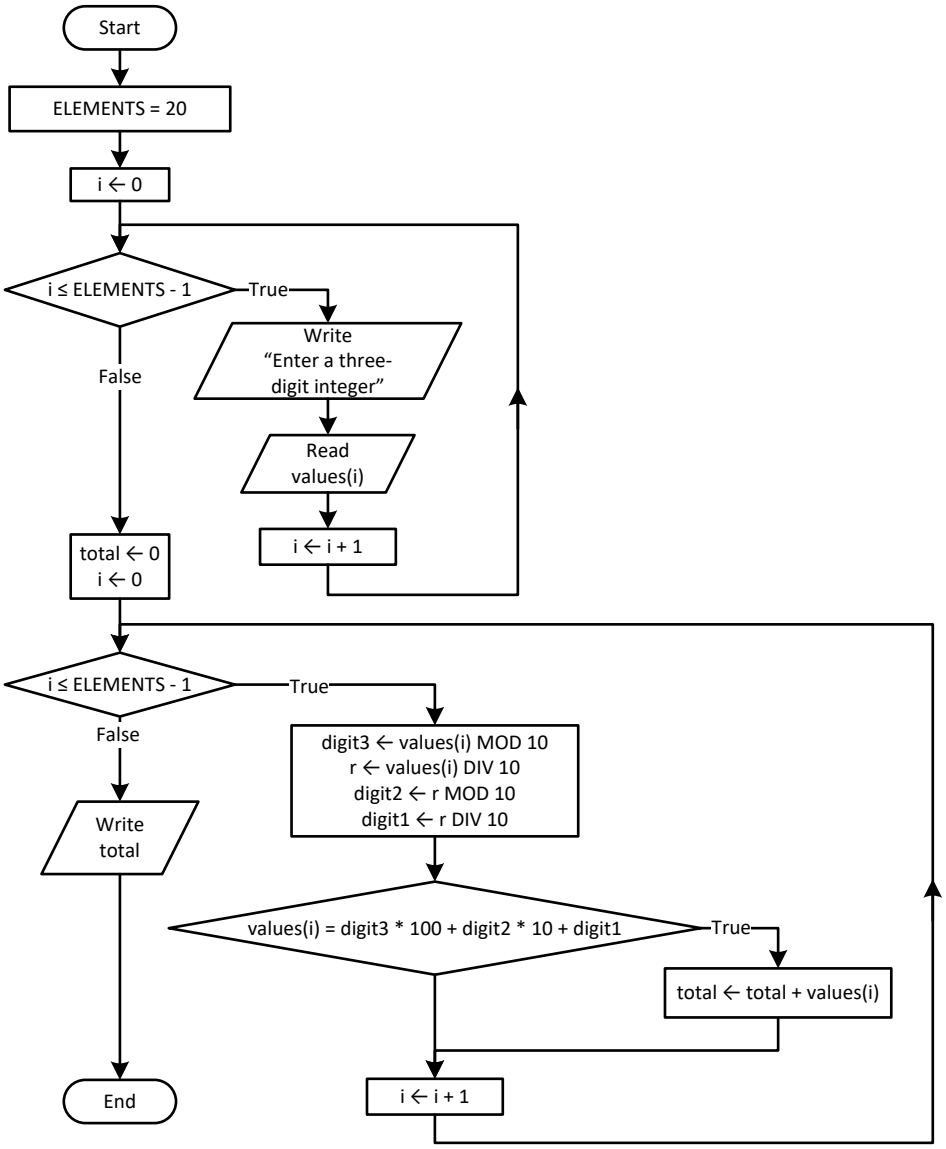
34.8 Review Exercises

1. Solution

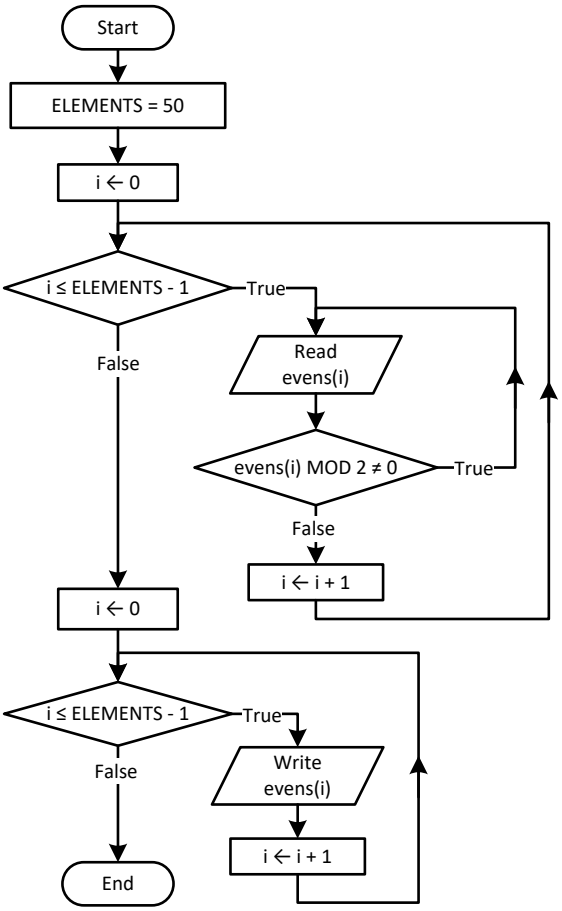


✎ *Please note that since flowcharts are a loose method to represent an algorithm, it is not necessary to initialize an array within a flowchart; that is, there is no need to represent the statement* `Dim values(ELEMENTS - 1) As Integer`.

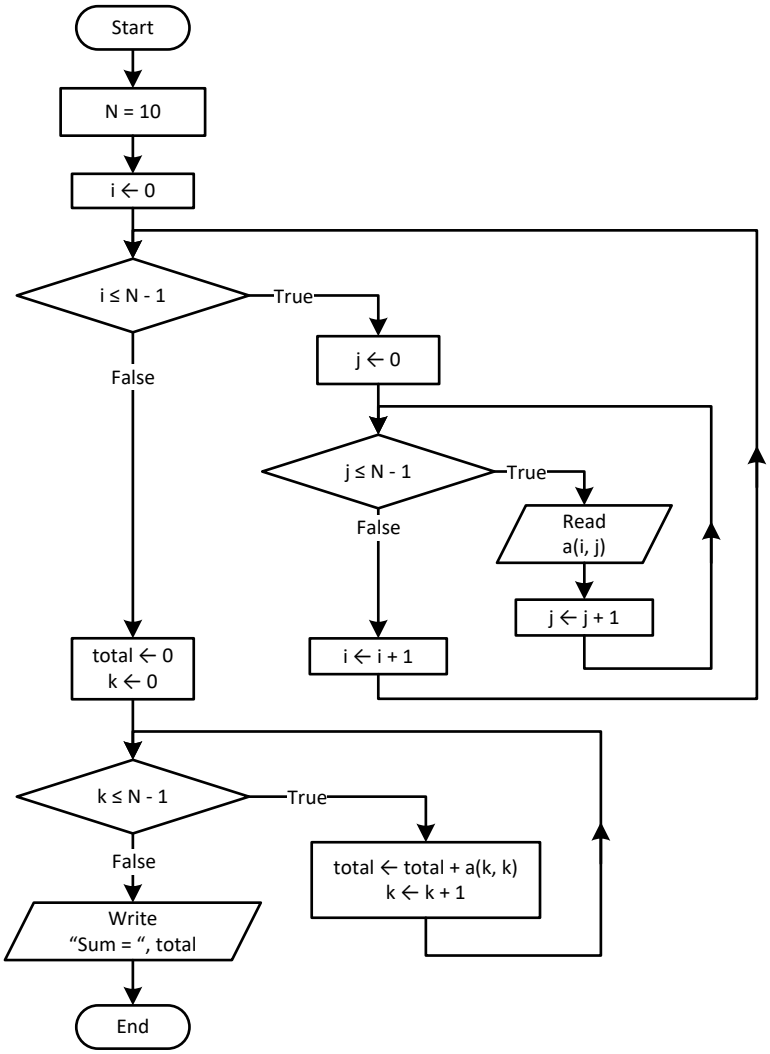
2. Solution


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

3. Solution


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

4. Solution



5. Solution

```
For i = 0 To CITIES - 1
  Do
    b(i) = Console.ReadLine()
  Loop While b(i) >= 0
Next
```

6. Solution

```
Sub Main(args As String())
  Dim i, m, n As Integer
  Dim b As Double
  Dim pos(89) As Double
  Dim neg(89) As Double

  i = 1
  m = 0
```

```
  n = 0
  Do
    b = Console.ReadLine()
    If b < 0 Then
      pos(m) = b
      m += 1
    Else
      neg(n) = b
      n += 1
    End If
    i += 1
  Loop While i < 90
  Console.WriteLine("The End")
End Sub
```

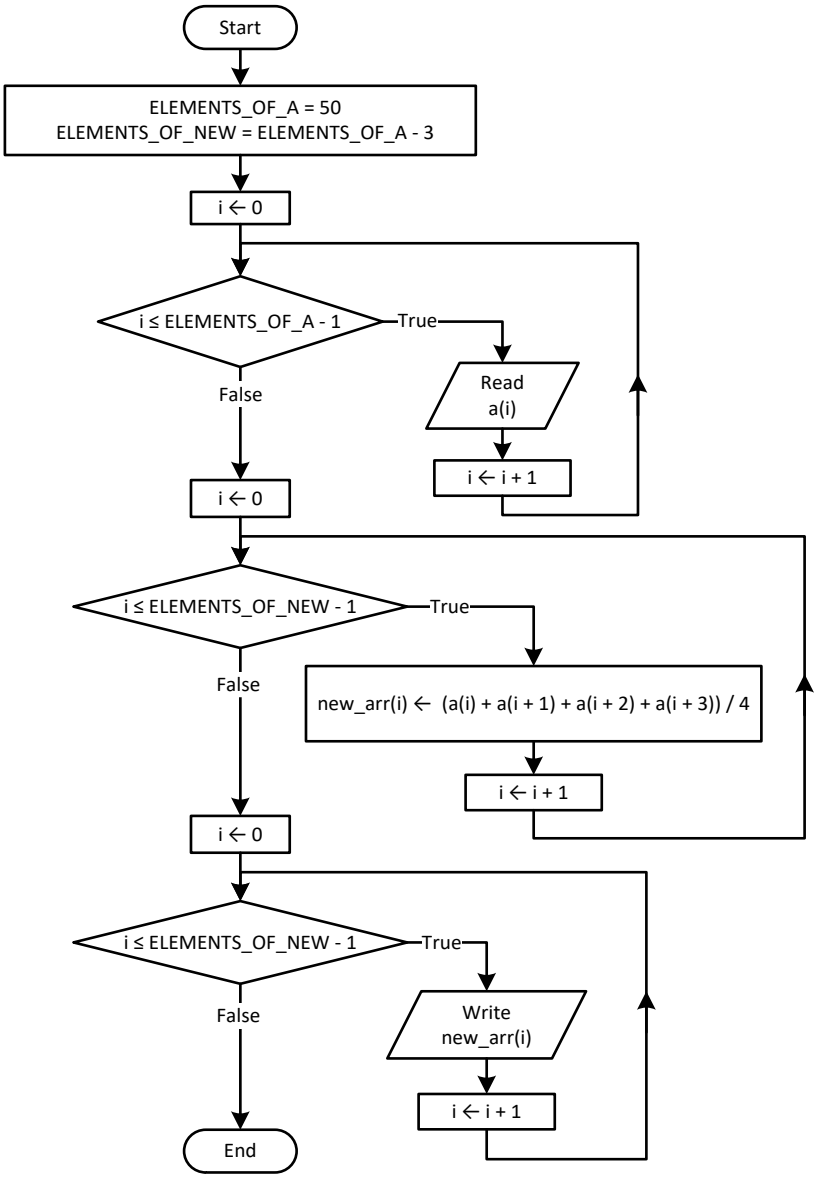
7. Solution

```
max_i = 0
max_j = 0
For i = 0 To CITIES - 1
  For j = 0 To CITIZENS - 1
    If a(i, j) > a(max_i, max_j) Then
      max_i = i
      max_j = j
    End If
  Next
Next

Console.WriteLine(a(max_i, max_j))
```

8. Solution

```
For i = 0 To ROWS - 1
  For j = 0 To COLUMNS - 1
    a(i, j) = Console.ReadLine()
    Do While a(i, j) = 0
      Console.WriteLine("Error")
      a(i, j) = Console.ReadLine()
    Loop
  Next
Next
```

9. Solution

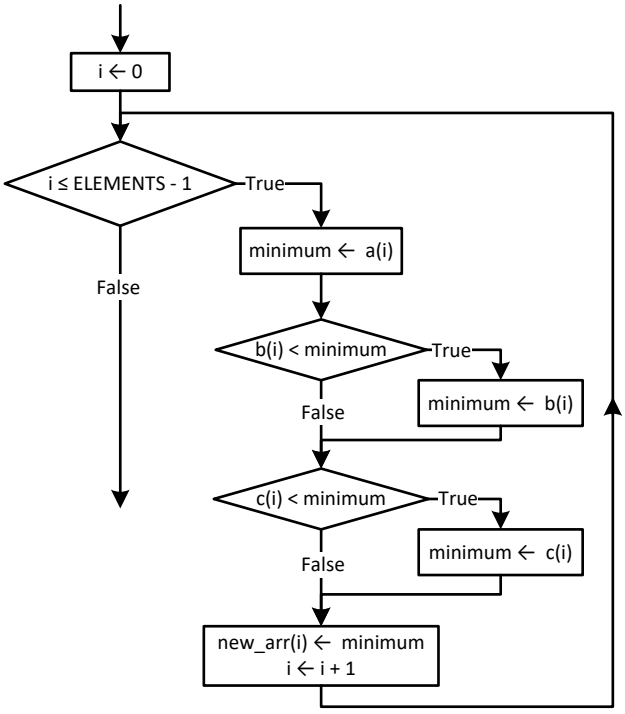



```
Const ELEMENTS_OF_A = 50
Const ELEMENTS_OF_NEW = ELEMENTS_OF_A - 3

Sub Main(args As String())
  Dim i As Integer

  Dim a(ELEMENTS_OF_A - 1) As Double
  For i = 0 To ELEMENTS_OF_A - 1
    a(i) = Console.ReadLine()
  Next

  Dim new_arr(ELEMENTS_OF_NEW - 1) As Double
  For i = 0 To ELEMENTS_OF_NEW - 1
    new_arr(i) = (a(i) + a(i + 1) + a(i + 2) + a(i + 3)) / 4
  Next
```

```
  For i = 0 To ELEMENTS_OF_NEW - 1
    Console.WriteLine(new_arr(i) & vbTab)
  Next
End Sub
```

10. Solution

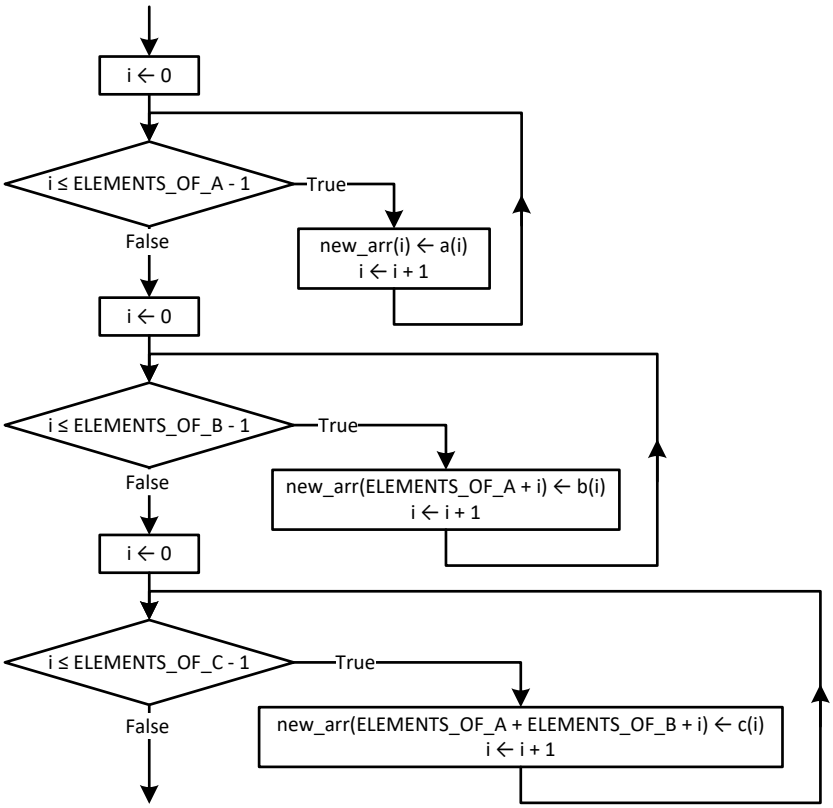
```
Const ELEMENTS = 15

Sub Main(args As String())
  Dim i As Integer
  Dim minimum

  Dim a(ELEMENTS - 1) As Double
  For i = 0 To ELEMENTS - 1
    a(i) = Console.ReadLine()
  Next

  Dim b(ELEMENTS - 1) As Double
  For i = 0 To ELEMENTS - 1
    b(i) = Console.ReadLine()
  Next

  Dim c(ELEMENTS - 1) As Double
  For i = 0 To ELEMENTS - 1
    c(i) = Console.ReadLine()
  Next

  Dim new_arr(ELEMENTS - 1) As Double
  For i = 0 To ELEMENTS - 1
    minimum = a(i)
    If b(i) < minimum Then
      minimum = b(i)
    End If
    If c(i) < minimum Then
      minimum = c(i)
    End If
    new_arr(i) = minimum
  Next

  For i = 0 To ELEMENTS - 1
    Console.WriteLine(new_arr(i))
  Next
End Sub
```



11. Solution

```
Const ELEMENTS_OF_A = 10
Const ELEMENTS_OF_B = 5
Const ELEMENTS_OF_C = 15
Const ELEMENTS_OF_NEW = ELEMENTS_OF_A + ELEMENTS_OF_B + ELEMENTS_OF_C

Sub Main(args As String())
  Dim i As Integer

  Dim a(ELEMENTS_OF_A - 1) As Double
  For i = 0 To ELEMENTS_OF_A - 1
    a(i) = Console.ReadLine()
  Next

  Dim b(ELEMENTS_OF_B - 1) As Double
  For i = 0 To ELEMENTS_OF_B - 1
    b(i) = Console.ReadLine()
  Next

  Dim c(ELEMENTS_OF_C - 1) As Double
  For i = 0 To ELEMENTS_OF_C - 1
    c(i) = Console.ReadLine()
  Next

  Dim new_arr(ELEMENTS_OF_NEW - 1) As Double
  For i = 0 To ELEMENTS_OF_C - 1
    new_arr(i) = c(i)
  Next
  For i = 0 To ELEMENTS_OF_B - 1
```

```
    new_arr(ELEMENTS_OF_C + i) = b(i)
  Next
  For i = 0 To ELEMENTS_OF_A - 1
    new_arr(ELEMENTS_OF_B + ELEMENTS_OF_C + i) = a(i)
  Next

  'Display array new
  For i = 0 To ELEMENTS_OF_NEW - 1
    Console.Write(new_arr(i) & vbTab)
  Next
End Sub
```



12. Solution

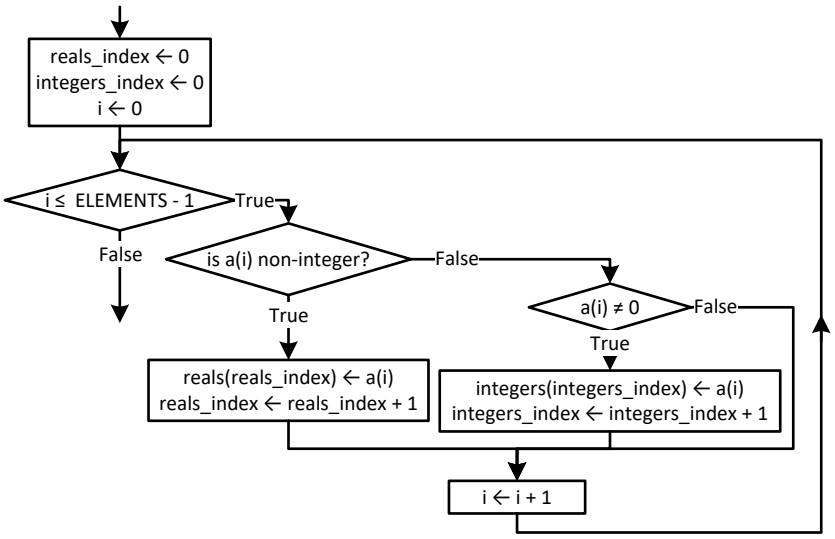
```
Const COLUMNS_OF_A = 10
Const COLUMNS_OF_B = 15
Const COLUMNS_OF_C = 20
Const ROWS = 5
Const COLUMNS = COLUMNS_OF_A + COLUMNS_OF_B + COLUMNS_OF_C

Sub Main(args As String())
  Dim i, j As Integer

  Dim a(ROWS - 1, COLUMNS_OF_A - 1) As Double
  For i = 0 To ROWS - 1
    For j = 0 To COLUMNS_OF_A - 1
      a(i, j) = Console.ReadLine()
```

```
    Next
  Next

  Dim b(ROWS - 1, COLUMNS_OF_B - 1) As Double
  For i = 0 To ROWS - 1
    For j = 0 To COLUMNS_OF_B - 1
      b(i, j) = Console.ReadLine()
    Next
  Next

  Dim c(ROWS - 1, COLUMNS_OF_C - 1) As Double
  For i = 0 To ROWS - 1
    For j = 0 To COLUMNS_OF_C - 1
      c(i, j) = Console.ReadLine()
    Next
  Next

  Dim new_arr(ROWS - 1, COLUMNS - 1) As Double
  For i = 0 To ROWS - 1
    For j = 0 To COLUMNS_OF_A - 1
      new_arr(i, j) = a(i, j)
    Next
  Next
  For i = 0 To ROWS - 1
    For j = 0 To COLUMNS_OF_B - 1
      new_arr(i, COLUMNS_OF_A + j) = b(i, j)
    Next
  Next
  For i = 0 To ROWS - 1
    For j = 0 To COLUMNS_OF_C - 1
      new_arr(i, COLUMNS_OF_A + COLUMNS_OF_B + j) = c(i, j)
    Next
  Next

  For i = 0 To ROWS - 1
    For j = 0 To COLUMNS - 1
      Console.Write(new_arr(i, j) & vbTab)
    Next
    Console.WriteLine()
  Next
End Sub
```

13. Solution

```
Const ELEMENTS = 50

Sub Main(args As String())
  Dim i, integers_index, reals_index As Integer

  Dim a(ELEMENTS - 1) As Double
  For i = 0 To ELEMENTS - 1
    a(i) = Console.ReadLine()
  Next
```

```
  Dim reals(ELEMENTS - 1) As Double
  Dim integers(ELEMENTS - 1) As Integer
  reals_index = 0
  integers_index = 0
  For i = 0 To ELEMENTS - 1
    If a(i) <> Fix(a(i)) Then
      reals(reals_index) = a(i)
      reals_index += 1
    ElseIf a(i) <> 0 Then
      integers(integers_index) = a(i)
      integers_index += 1
    End If
  Next

  For i = 0 To reals_index - 1
    Console.Write(reals(i) & vbTab)
  Next

  Console.WriteLine()
  For i = 0 To integers_index - 1
    Console.Write(integers(i) & vbTab)
  Next
End Sub
```



14. Solution

```
Const ELEMENTS = 50

Sub Main(args As String())
  Dim digit1, digit2, digit3, i, k, r As Integer

  Dim a(ELEMENTS - 1) As Integer
  For i = 0 To ELEMENTS - 1
    a(i) = Console.ReadLine()
  Next
```

```
  Dim b(ELEMENTS - 1) As Integer
  k = 0
  For i = 0 To ELEMENTS - 1
    digit3 = a(i) Mod 10
    r = a(i) \ 10
    digit2 = r Mod 10
    digit1 = r \ 10

    If digit1 < digit2 And digit2 < digit3 Then
      b(k) = a(i)
      k += 1
    End If
  Next

  For i = 0 To k - 1
    Console.Write(b(i) & vbTab)
  Next
End Sub
```

15. Solution

```
Const PRODUCTS = 10
Const CITIZENS = 200

Sub Main(args As String())
  Dim count_B, i, j, maximum As Integer

  Dim prod_names(PRODUCTS - 1) As String
  Dim answers(PRODUCTS - 1, CITIZENS - 1) As String
  For i = 0 To PRODUCTS - 1
    prod_names(i) = Console.ReadLine()
    For j = 0 To CITIZENS - 1
      answers(i, j) = Console.ReadLine()
      Do While answers(i, j).CompareTo("A") < 0 Or answers(i, j).CompareTo("D") > 0
        Console.WriteLine("Error! ")
        answers(i, j) = Console.ReadLine()
      Loop
    Next
  Next

  Dim count_A(PRODUCTS - 1) As Integer
  For i = 0 To PRODUCTS - 1
    count_A(i) = 0
    For j = 0 To CITIZENS - 1
      If answers(i, j) = "A" Then
        count_A(i) += 1
      End If
    Next
    Console.WriteLine(prod_names(i) & ", " & count_A(i))
  Next

  For j = 0 To CITIZENS - 1
```

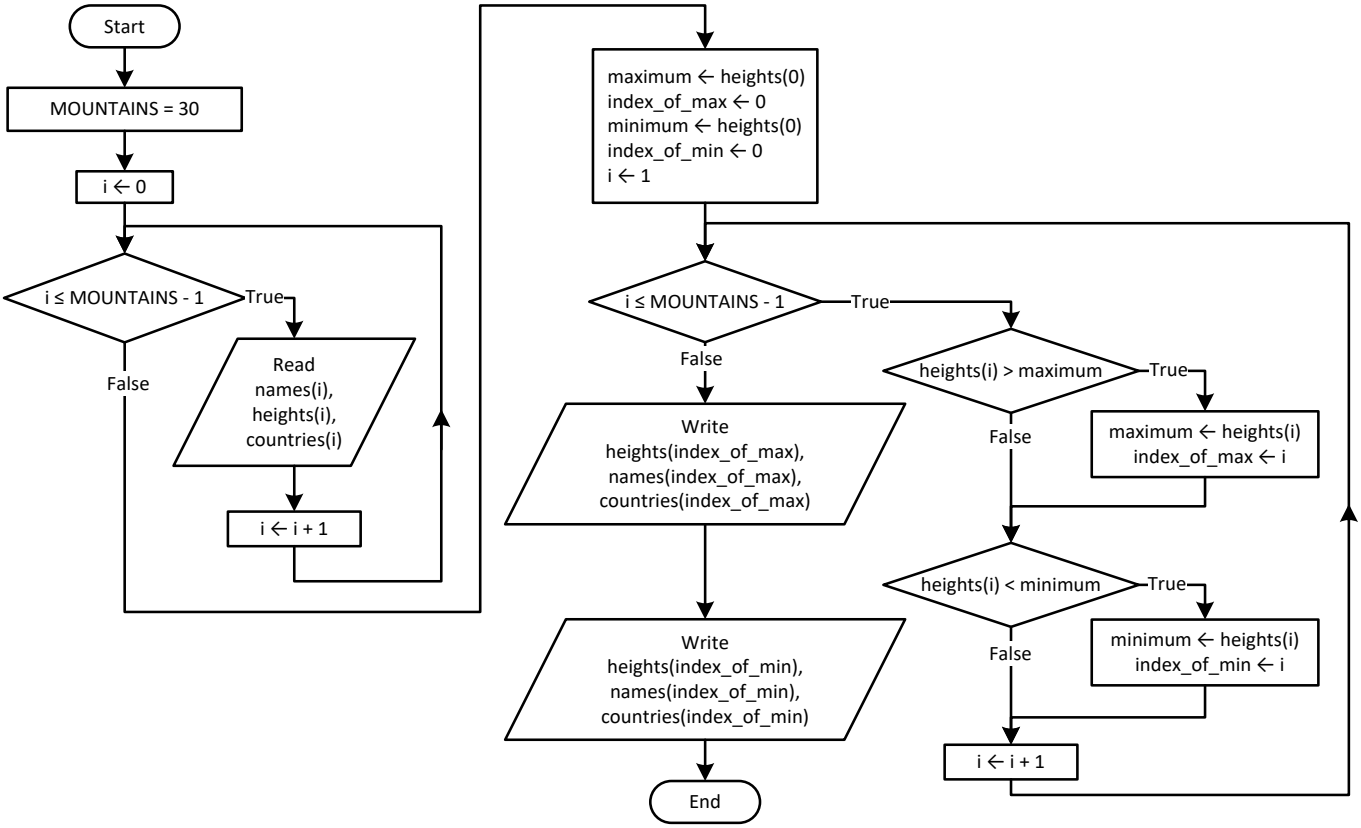
```
    count_B = 0
    For i = 0 To PRODUCTS - 1
      If answers(i, j) = "B" Then
        count_B += 1
      End If
    Next
    Console.WriteLine(count_B)
  Next

  maximum = count_A(0)
  For i = 1 To PRODUCTS - 1
    If count_A(i) > maximum Then
      maximum = count_A(i)
    End If
  Next

  For i = 0 To PRODUCTS - 1
    If count_A(i) = maximum Then
      Console.WriteLine(prod_names(i))
    End If
  Next
End Sub
```

16. Solution

```
Const US_CITIES = 20
Const CANADIAN_CITIES = 20

Sub Main(args As String())
  Dim i, j, min_j As Integer
  Dim minimum As Double

  Dim us_names(US_CITIES - 1) As String
  For i = 0 To US_CITIES - 1
    Console.WriteLine("Enter name for US city No " & (i + 1) & ": ")
    us_names(i) = Console.ReadLine()
  Next

  Dim canadian_names(CANADIAN_CITIES - 1) As String
  For j = 0 To CANADIAN_CITIES - 1
    Console.WriteLine("Enter name for Canadian city No " & (j + 1) & ": ")
    canadian_names(j) = Console.ReadLine()
  Next

  Dim distances(US_CITIES - 1, CANADIAN_CITIES - 1) As Double
  For i = 0 To US_CITIES - 1
    For j = 0 To CANADIAN_CITIES - 1
      Console.WriteLine("Enter distance between " & us_names(i) & " and " & canadian_names(j) & ": ")
      distances(i, j) = Console.ReadLine()
    Next
  Next

  For i = 0 To US_CITIES - 1
```

```
    minimum = distances(i, 0)
    min_j = 0
    For j = 1 To CANADIAN_CITIES - 1
      If distances(i, j) < minimum Then
        minimum = distances(i, j)
        min_j = j
      End If
    Next
    Console.WriteLine("Closest Canadian city to " & us_names(i) & " is " & canadian_names(min_j))
  Next
End Sub
```

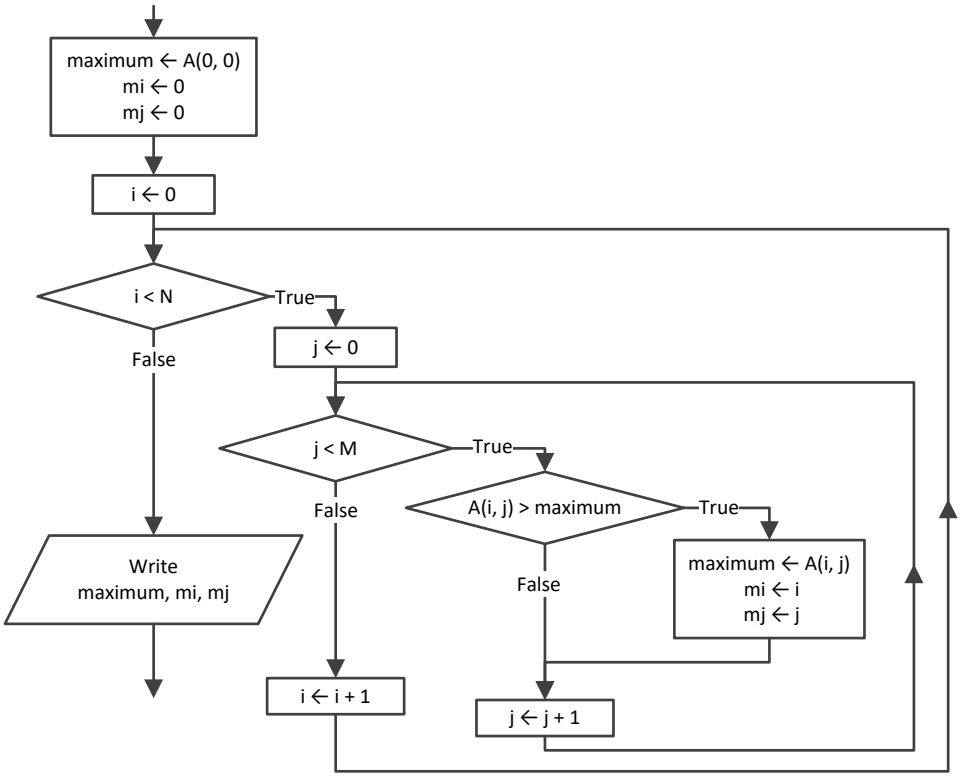
17. Solution




```
Const MOUNTAINS = 30

Sub Main(args As String())
  Dim i, index_of_max, index_of_min As Integer
  Dim maximum, minimum As Double

  Dim names(MOUNTAINS - 1) As String
  Dim heights(MOUNTAINS - 1) As Double
  Dim countries(MOUNTAINS - 1) As String
  For i = 0 To MOUNTAINS - 1
    names(i) = Console.ReadLine()
    heights(i) = Console.ReadLine()
    countries(i) = Console.ReadLine()
```

```
  Next

  maximum = heights(0)
  index_of_max = 0
  minimum = heights(0)
  index_of_min = 0
  For i = 1 To MOUNTAINS - 1
    If heights(i) > maximum Then
      maximum = heights(i)
      index_of_max = i
    End If
    If heights(i) < minimum Then
      minimum = heights(i)
      index_of_min = i
    End If
  Next

  Console.WriteLine(heights(index_of_max) & ", " & names(index_of_max) & ", " & countries(index_of_max))
  Console.WriteLine(heights(index_of_min) & ", " & names(index_of_min) & ", " & countries(index_of_min))
End Sub
```

18. Solution



19. Solution

```
Const TEAMS = 26
Const GAMES = 15
```

```
Sub Main(args As String())
  Dim i, j, m_i, maximum As Integer

  Dim names(TEAMS - 1) As String
  Dim results(TEAMS - 1, GAMES - 1) As String
  For i = 0 To TEAMS - 1
    names(i) = Console.ReadLine()
    For j = 0 To GAMES - 1
      results(i, j) = Console.ReadLine()
    Next
  Next

  Dim points(TEAMS - 1) As Integer
  For i = 0 To TEAMS - 1
    points(i) = 0
    For j = 0 To GAMES - 1
      If results(i, j) = "W" Then
        points(i) += 3
      ElseIf results(i, j) = "T" Then
        points(i) += 1
      End If
    Next
  Next

  maximum = points(0)
  m_i = 0
  For i = 1 To TEAMS - 1
    If points(i) > maximum Then
      maximum = points(i)
      m_i = i
    End If
  Next

  Console.WriteLine(names(m_i))
End Sub
```

20. Solution

```
Const OBJECTS = 10
Const FALLS = 20

Sub Main(args As String())
  Dim i, j As Integer
  Dim maxi, mini As Double

  Dim heights(OBJECTS - 1, FALLS - 1) As Double
  Dim times(OBJECTS - 1, FALLS - 1) As Double
  For i = 0 To OBJECTS - 1
    For j = 0 To FALLS - 1
      heights(i, j) = Console.ReadLine()
      times(i, j) = Console.ReadLine()
    Next
  Next
```

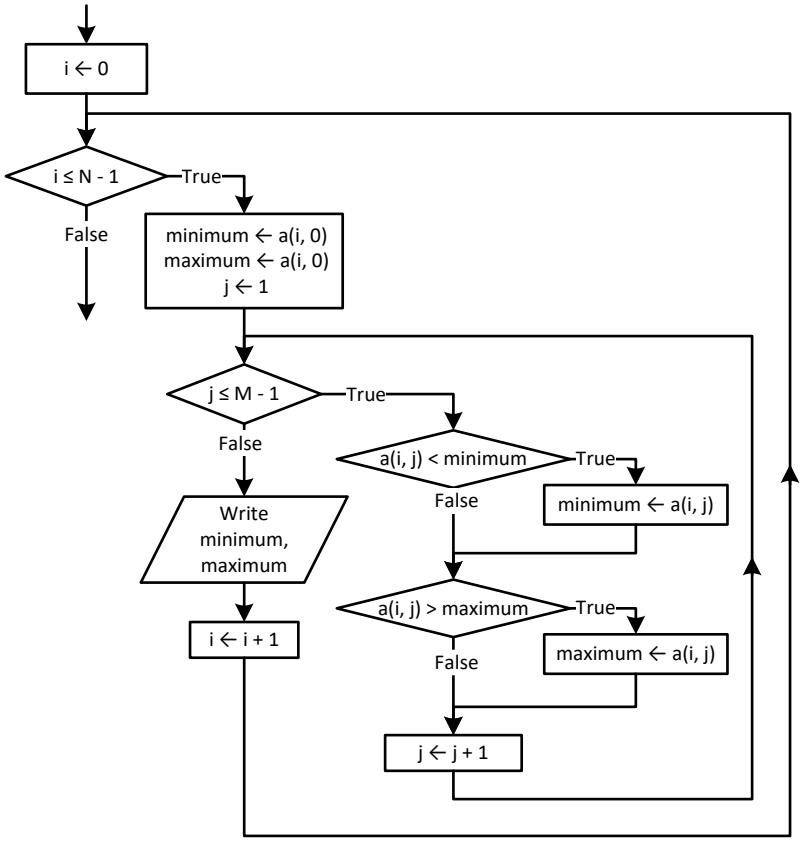
```
  Dim g(OBJECTS - 1, FALLS - 1) As Double
  For i = 0 To OBJECTS - 1
    For j = 0 To FALLS - 1
      g(i, j) = 2 * heights(i, j) / times(i, j) ^ 2
    Next
  Next

  Dim minimum(OBJECTS - 1) As Double
  Dim maximum(OBJECTS - 1) As Double
  For i = 0 To OBJECTS - 1
    minimum(i) = g(i, 0)
    maximum(i) = g(i, 0)
    For j = 1 To FALLS - 1
      If g(i, j) < minimum(i) Then
        minimum(i) = g(i, j)
      End If
      If g(i, j) > maximum(i) Then
        maximum(i) = g(i, j)
      End If
    Next
  Next

  For i = 0 To OBJECTS - 1
    Console.WriteLine(minimum(i) & ", " & maximum(i))
  Next

  maxi = maximum(0)
  mini = minimum(0)
  For i = 1 To OBJECTS - 1
    If maximum(i) > maxi Then
      maxi = maximum(i)
    End If
    If minimum(i) < mini Then
      mini = minimum(i)
    End If
  Next

  Console.WriteLine(mini & ", " & maxi)
End Sub
```

21. Solution

```
Const STATIONS = 10
Const DAYS = 365

Sub Main(args As String())
  Dim i, j, m_i As Integer
  Dim minimum As Double

  Dim names(STATIONS - 1) As String
  Dim co2(STATIONS - 1, DAYS - 1) As Double
  For i = 0 To STATIONS - 1
```

```
    names(i) = Console.ReadLine()
    For j = 0 To DAYS - 1
      co2(i, j) = Console.ReadLine()
    Next
  Next

  Dim average(STATIONS - 1) As Double
  For i = 0 To STATIONS - 1
    average(i) = 0
    For j = 0 To DAYS - 1
      average(i) += co2(i, j)
    Next
    average(i) /= DAYS
  Next

  minimum = average(0)
  m_i = 0
  For i = 1 To STATIONS - 1
    If average(i) < minimum Then
      minimum = average(i)
      m_i = i
    End If
  Next
  Console.WriteLine(names(m_i))
End Sub
```

22. Solution



23. Solution

First approach

```
Const ROWS = 20
Const COLUMNS = 30

Sub Main(args As String())
  Dim i, j As Integer

  Dim b(ROWS - 1, COLUMNS - 1) As Double
  For i = 0 To ROWS - 1
    For j = 0 To COLUMNS - 1
      b(i, j) = Console.ReadLine()
    Next
  Next

  Dim minimum(COLUMNS - 1) As Double
  Dim maximum(COLUMNS - 1) As Double
  For j = 0 To COLUMNS - 1
    minimum(j) = b(0, j)
    maximum(j) = b(0, j)
    For i = 1 To ROWS - 1
      If b(i, j) < minimum(j) Then
```

```
        minimum(j) = b(i, j)
      End If
      If b(i, j) > maximum(j) Then
        maximum(j) = b(i, j)
      End If
    Next
  Next

  For j = 0 To COLUMNS - 1
    Console.WriteLine(minimum(j) & " " & maximum(j))
  Next
End Sub
```

Second approach

```
Const ROWS = 20
Const COLUMNS = 30

Sub Main(args As String())
  Dim i, j As Integer
  Dim minimum, maximum As Double

  Dim b(ROWS - 1, COLUMNS - 1) As Double
  For i = 0 To ROWS - 1
    For j = 0 To COLUMNS - 1
      b(i, j) = Console.ReadLine()
    Next
  Next

  For j = 0 To COLUMNS - 1
    minimum = b(0, j)
    maximum = b(0, j)
    For i = 1 To ROWS - 1
      If b(i, j) < minimum Then
        minimum = b(i, j)
      End If
      If b(i, j) > maximum Then
        maximum = b(i, j)
      End If
    Next
    Console.WriteLine(minimum & " " & maximum)
  Next
End Sub
```

24. Solution

```
Const TEAMS = 20
Const GAMES = 10

Sub Main(args As String())
  Dim i, j, m, n, temp As Integer
  Dim swaps As Boolean
  Dim temp_str As String
```

```
  Dim names(TEAMS - 1) As String
  Dim results(TEAMS - 1, GAMES - 1) As String
  For i = 0 To TEAMS - 1
    Console.Write("Enter team name: ")
    names(i) = Console.ReadLine()
    For j = 0 To GAMES - 1
      Console.WriteLine("Enter result for team " & names(i) & " for game No " & (j + 1) & ": ")
      results(i, j) = Console.ReadLine()
      Do While results(i, j) <> "W" And results(i, j) <> "L" And results(i, j) <> "T"
        Console.Write("Error! Enter only value W, L, or T: ")
        results(i, j) = Console.ReadLine()
      Loop
    Next
  Next

  Dim points(TEAMS - 1) As Integer
  For i = 0 To TEAMS - 1
    points(i) = 0
    For j = 0 To GAMES - 1
      If results(i, j) = "W" Then
        points(i) += 3
      ElseIf results(i, j) = "T" Then
        points(i) += 1
      End If
    Next
  Next

  For m = 1 To TEAMS - 1
    swaps = False
    For n = TEAMS - 1 To m Step -1
      If points(n) > points(n - 1) Then
        temp = points(n)
        points(n) = points(n - 1)
        points(n - 1) = temp

        temp_str = names(n)
        names(n) = names(n - 1)
        names(n - 1) = temp_str

        swaps = True
      End If
    Next
    If Not swaps Then Exit For
  Next

  Console.WriteLine("Gold: " & names(0))
  Console.WriteLine("Silver: " & names(1))
  Console.WriteLine("Bronze: " & names(2))
End Sub
```

25. Solution

```
Const PEOPLE = 50

Sub Main(args As String())
  Dim i, m, n As Integer
  Dim temp As Double
  Dim temp_str As String

  Dim names(PEOPLE - 1) As String
  Dim heights(PEOPLE - 1) As Double
  For i = 0 To PEOPLE - 1
    Console.Write("Enter name for person No. " & (i + 1) & ": ")
    names(i) = Console.ReadLine()
    Console.Write("Enter height for person No. " & (i + 1) & ": ")
    heights(i) = Console.ReadLine()
  Next

  For m = 1 To PEOPLE - 1
    For n = PEOPLE - 1 To m Step -1
      If heights(n) > heights(n - 1) Then
        temp = heights(n)
        heights(n) = heights(n - 1)
        heights(n - 1) = temp

        temp_str = names(n)
        names(n) = names(n - 1)
        names(n - 1) = temp_str
      ElseIf heights(n) = heights(n - 1) Then
        If names(n).CompareTo(names(n - 1)) < 0 Then
          temp_str = names(n)
          names(n) = names(n - 1)
          names(n - 1) = temp_str
        End If
      End If
    Next
  Next

  For i = 0 To PEOPLE - 1
    Console.WriteLine(heights(i) & vbTab & names(i))
  Next
End Sub
```

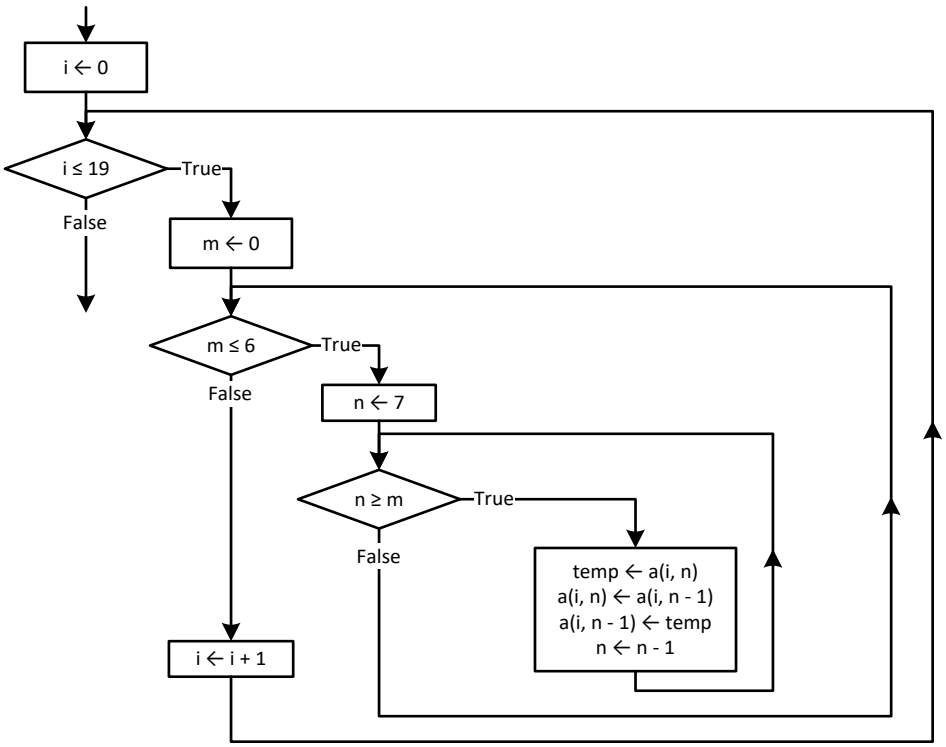
26. Solution

```
Const ARTISTS = 12
Const JUDGES = 10

Sub Main(args As String())
  Dim i, j, m, maximum, minimum, n, temp As Integer
  Dim temp_str As String

  Dim artist_names(ARTISTS - 1) As String
```

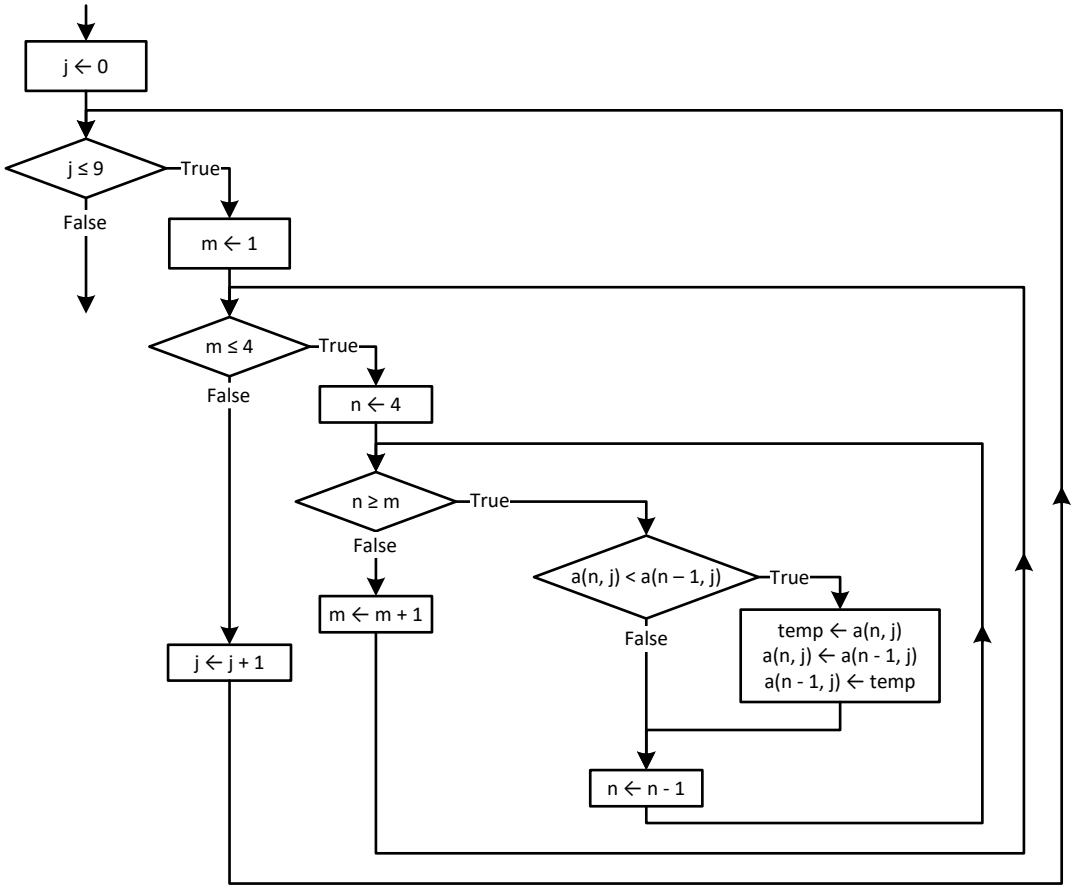
```
Dim score(ARTISTS - 1, JUDGES - 1) As Integer
For i = 0 To ARTISTS - 1
  Console.WriteLine("Enter name for artist No " & (i + 1) & ": ")
  artist_names(i) = Console.ReadLine()
  For j = 0 To JUDGES - 1
    Console.Write("Enter score for artist: " & artist_names(i))
    Console.WriteLine(" gotten from judge No " & (j + 1) & ": ")
    score(i, j) = Console.ReadLine()
  Next
Next

Dim total(ARTISTS - 1) As Integer
For i = 0 To ARTISTS - 1
  total(i) = 0
  For j = 1 To JUDGES - 1
    total(i) += score(i, j)
  Next
Next

For i = 0 To ARTISTS - 1
  minimum = score(i, 0)
  maximum = score(i, 0)
  For j = 1 To JUDGES - 1
    If score(i, j) < minimum Then
      minimum = score(i, j)
    End If
    If score(i, j) > maximum Then
      maximum = score(i, j)
    End If
  Next
  total(i) = total(i) - minimum - maximum
  Console.WriteLine(total(i))
Next

For m = 1 To ARTISTS - 1
  For n = ARTISTS - 1 To m Step -1
    If total(n) > total(n - 1) Then
      temp = total(n)
      total(n) = total(n - 1)
      total(n - 1) = temp

      temp_str = artist_names(n)
      artist_names(n) = artist_names(n - 1)
      artist_names(n - 1) = temp_str
    ElseIf total(n) = total(n - 1) Then
      If artist_names(n).CompareTo(artist_names(n - 1)) < 0 Then
        temp_str = artist_names(n)
        artist_names(n) = artist_names(n - 1)
        artist_names(n - 1) = temp_str
      End If
    End If
  Next
```

```
  Next

  For i = 0 To ARTISTS - 1
    Console.WriteLine(artist_names(i) & ", " & total(i))
  Next
End Sub
```

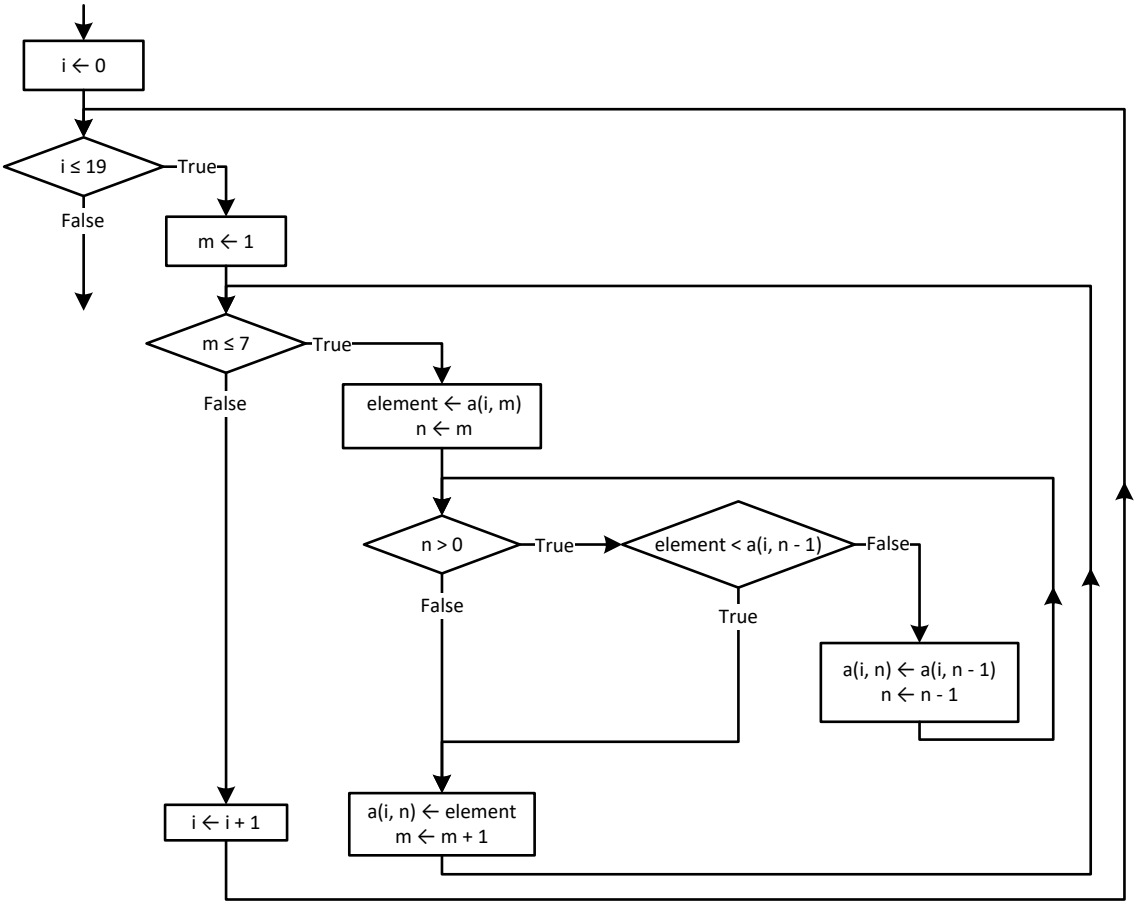
27. Solution



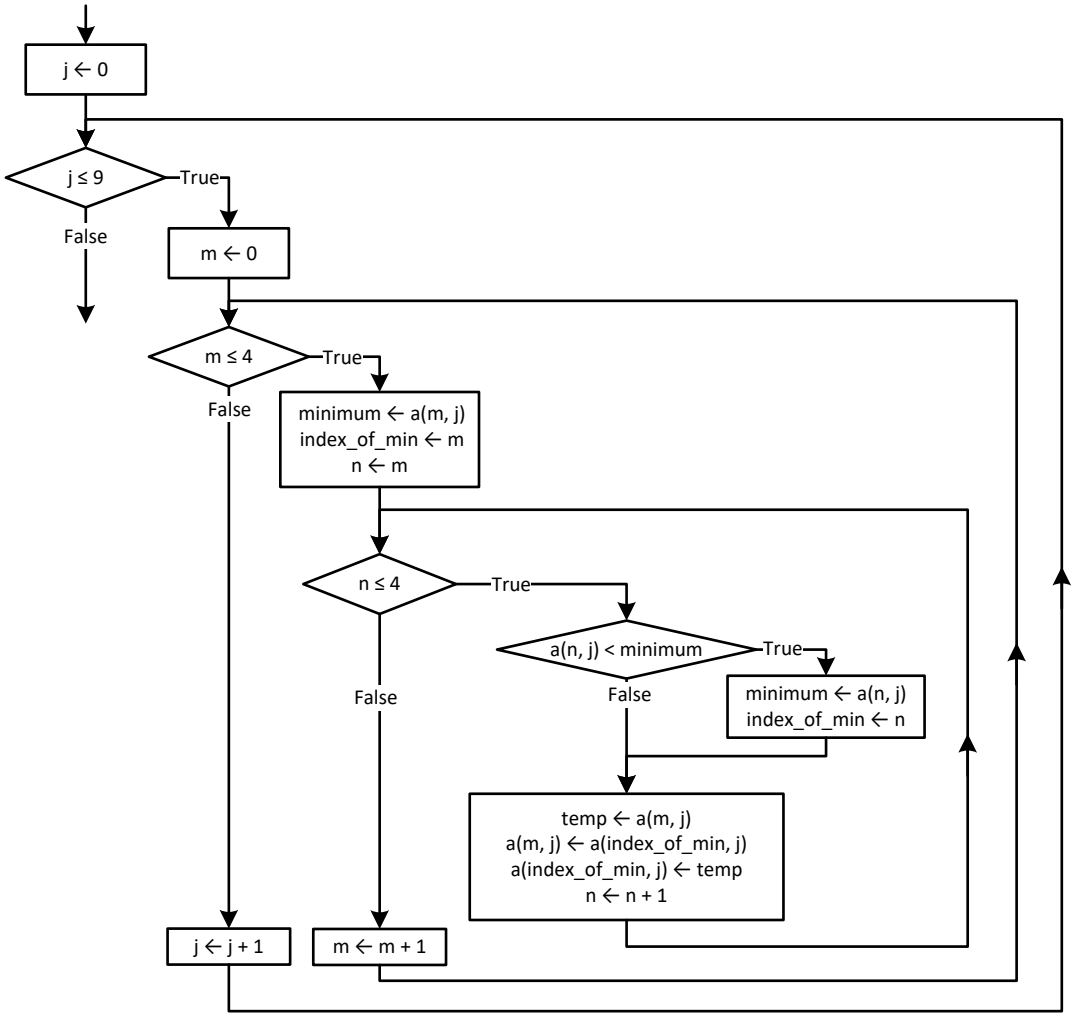
28. Solution


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
m ← m + 1
j ← j + 1
temp ← a(n, j)
a(n, j) ← a(n - 1, j)
a(n - 1, j) ← temp
n ← n - 1

29. Solution


i ← 0
i ≤ 19
True
False
m ← 1
m ≤ 7
True
False
element ← a(i, m)
n ← m
n > 0
True
False
element < a(i, n - 1)
False
True
a(i, n) ← a(i, n - 1)
n ← n - 1
i ← i + 1
a(i, n) ← element
m ← m + 1

30. Solution



31. Solution

```
Const PEOPLE = 10
Const PUZZLES = 8

Sub Main(args As String())
  Dim i, index_of_min, j, m, n As Integer
  Dim minimum, temp As Double
  Dim temp_str As String

  Dim names(PEOPLE - 1) As String
  Dim times(PEOPLE - 1, PUZZLES - 1) As Double
  For i = 0 To PEOPLE - 1
    names(i) = Console.ReadLine()
    For j = 0 To PUZZLES - 1
      times(i, j) = Console.ReadLine()
    Next
  Next
```

```
For i = 0 To PEOPLE - 1
  For m = 0 To PUZZLES - 1
    minimum = times(i, m)
    index_of_min = m
    For n = m To PUZZLES - 1
      If times(i, n) < minimum Then
        minimum = times(i, n)
        index_of_min = n
      End If
    Next
    temp = times(i, m)
    times(i, m) = times(i, index_of_min)
    times(i, index_of_min) = temp
  Next
Next

For i = 0 To PEOPLE - 1
  Console.WriteLine(names(i))
  For j = 0 To 2
    Console.WriteLine(times(i, j))
  Next
Next

Dim average(PEOPLE - 1) As Double
For i = 0 To PEOPLE - 1
  average(i) = 0
  For j = 0 To PUZZLES - 1
    average(i) += times(i, j)
  Next
  average(i) /= PUZZLES
Next

For m = 0 To PEOPLE - 1
  minimum = average(m)
  index_of_min = m
  For n = m To PEOPLE - 1
    If average(n) < minimum Then
      minimum = average(n)
      index_of_min = n
    End If
  Next
  temp = average(m)
  average(m) = average(index_of_min)
  average(index_of_min) = temp

  temp_str = names(m)
  names(m) = names(index_of_min)
  names(index_of_min) = temp_str
Next

Console.WriteLine(names(0) & ", " & names(1) & ", " & names(2))
```

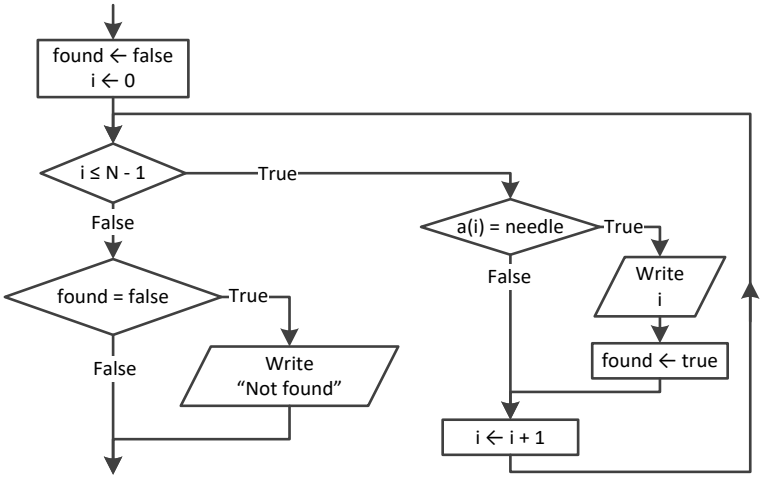
```
End Sub
```

32. Solution

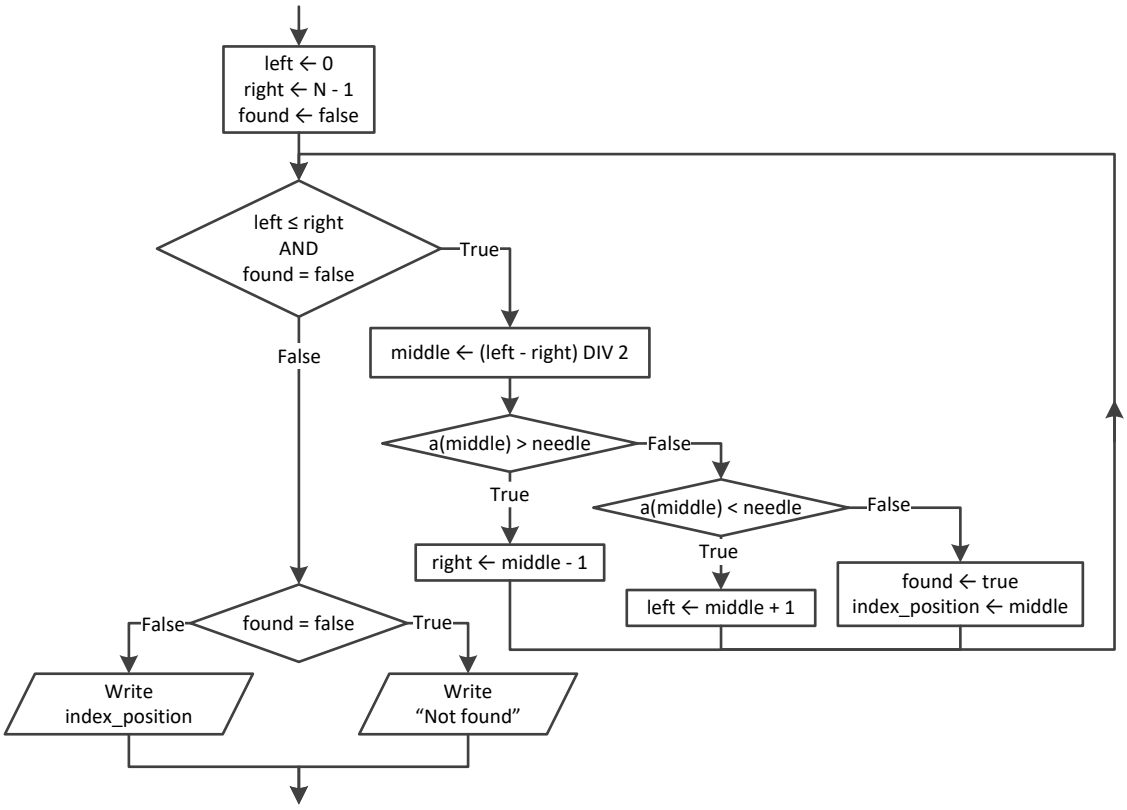
```
Const AREAS = 5
Const HOURS = 48

Sub Main(args As String())
  Dim i, j, m, m_i, m_j, n As Integer
  Dim maximum, element_1 As Double
  Dim element_2 As String

  Dim names(AREAS - 1) As String
  Dim CO2(AREAS - 1, HOURS - 1) As Double
  For i = 0 To AREAS - 1
    names(i) = Console.ReadLine()
    For j = 0 To HOURS - 1
      CO2(i, j) = Console.ReadLine()
    Next
  Next

  Dim average_per_hour(AREAS - 1) As Double
  For i = 0 To AREAS - 1
    average_per_hour(i) = 0
    For j = 0 To HOURS - 1
      average_per_hour(i) += CO2(i, j)
    Next
    average_per_hour(i) /= HOURS
  Next

  For i = 0 To AREAS - 1
    Console.WriteLine(names(i) & ", " & average_per_hour(i))
  Next

  Dim average_per_city(HOURS - 1) As Double
  For j = 0 To HOURS - 1
    average_per_city(j) = 0
    For i = 0 To AREAS - 1
      average_per_city(j) += CO2(i, j)
    Next
    average_per_city(j) /= AREAS
  Next

  For j = 0 To HOURS - 1
    Console.WriteLine(average_per_city(j))
  Next

  maximum = average_per_city(0)
  m_j = 0
  For j = 1 To HOURS - 1
    If average_per_city(j) > maximum Then
      maximum = average_per_city(j)
      m_j = j
```

```
    End If
  Next
  Console.WriteLine(m_j)

  maximum = CO2(0, 0)
  m_i = 0
  m_j = 0
  For i = 0 To AREAS - 1
    For j = 0 To HOURS - 1
      If CO2(i, j) > maximum Then
        maximum = CO2(i, j)
        m_i = i
        m_j = j
      End If
    Next
  Next
  Console.WriteLine(m_j & ", " & names(m_i))

  For m = 1 To AREAS - 1
    element_1 = average_per_hour(m)
    element_2 = names(m)

    n = m
    Do While n > 0
      If element_1 < average_per_hour(n - 1) Then Exit Do
      average_per_hour(n) = average_per_hour(n - 1)
      names(n) = names(n - 1)
      n -= 1
    Loop

    average_per_hour(n) = element_1
    names(n) = element_2
  Next

  Console.WriteLine(names(0) & ", " & names(1)  & ", " &  names(2))
End Sub
```

33. Solution



34. Solution



35. Solution

```
Const TEAMS = 10
Const GAMES = 16

Sub Main(args As String())
```

```
  Dim i, j, total As Integer
  Dim needle As String

  Dim names(TEAMS - 1) As String
  Dim goals_scored(TEAMS - 1, GAMES - 1) As Integer
  Dim goals_let_in(TEAMS - 1, GAMES - 1) As Integer
  For i = 0 To TEAMS - 1
    Console.Write("Enter team name: ")
    names(i) = Console.ReadLine()
    For j = 0 To GAMES - 1
      Console.Write("Enter goals scored: ")
      goals_scored(i, j) = Console.ReadLine()
      Do While goals_scored(i, j) < 0
        Console.Write("Error! Enter goals scored: ")
        goals_scored(i, j) = Console.ReadLine()
      Loop

      Console.Write("Enter goals let in: ")
      goals_let_in(i, j) = Console.ReadLine()
      Do While goals_let_in(i, j) < 0
        Console.Write("Error! Enter goals let in: ")
        goals_let_in(i, j) = Console.ReadLine()
      Loop
    Next
  Next

  Console.Write("Enter a team to search: ")
  needle = Console.ReadLine()

  i = 0
  Do While i < TEAMS - 1 And names(i) <> needle
    i += 1
  Loop

  If names(i) <> needle Then
    Console.WriteLine("This team does not exist")
  Else
    total = 0
    For j = 0 To GAMES - 1
      If goals_scored(i, j) > goals_let_in(i, j) Then
        total += 3
      ElseIf goals_scored(i, j) = goals_let_in(i, j) Then
        total += 1
      End If
    Next
    Console.WriteLine(total)
  End If
End Sub
```

36. Solution

```
Const CLASS1 = 20
```

```
Const CLASS2 = 25

Sub Main(args As String())
  Dim i, left, m, middle, n, right As Integer
  Dim temp, needle As String
  Dim found As Boolean

  Console.WriteLine("Class 1")
  Dim names1(CLASS1 - 1) As String
  For i = 0 To CLASS1 - 1
    Console.Write("Enter name: ")
    names1(i) = Console.ReadLine()
  Next
  Console.WriteLine("Class 2")
  Dim names2(CLASS2 - 1) As String
  For i = 0 To CLASS2 - 1
    Console.Write("Enter name: ")
    names2(i) = Console.ReadLine()
  Next

  'Bubble sort
  For m = 1 To CLASS1 - 1
    For n = CLASS1 - 1 To m Step -1
      If names1(n).CompareTo(names1(n - 1)) < 0 Then
        temp = names1(n)
        names1(n) = names1(n - 1)
        names1(n) = temp
      End If
    Next
  Next
  For m = 1 To CLASS2 - 1
    For n = CLASS2 - 1 To m Step -1
      If names2(n).CompareTo(names2(n - 1)) < 0 Then
        temp = names2(n)
        names2(n) = names2(n - 1)
        names2(n) = temp
      End If
    Next
  Next

  Console.WriteLine(vbCrLf & "Class 1")
  For i = 0 To CLASS1 - 1
    Console.WriteLine(names1(i))
  Next
  Console.WriteLine(vbCrLf & "Class 2")
  For i = 0 To CLASS2 - 1
    Console.WriteLine(names2(i))
  Next

  Console.Write("Enter a name to search: ")
  needle = Console.ReadLine()
```

```
  left = 0
  right = CLASS1 - 1
  found = False
  Do While left <= right And Not found
    middle = (left + right) \ 2

    If names1(middle).CompareTo(needle) > 0 Then
      right = middle - 1
    ElseIf names1(middle).CompareTo(needle) < 0 Then
      left = middle + 1
    Else
      found = True
    End If
  Loop

  If found Then
    Console.WriteLine("Student found in Class No 1")
  Else
    left = 0
    right = CLASS2 - 1
    Do While left <= right And Not found
      middle = (left + right) \ 2

      If names2(middle).CompareTo(needle) > 0 Then
        right = middle - 1
      ElseIf names2(middle).CompareTo(needle) < 0 Then
        left = middle + 1
      Else
        found = True
      End If
    Loop

    If found Then
      Console.WriteLine("Student found in Class No 2")
    Else
      Console.WriteLine("Student not found in either class")
    End If
  End If
End Sub
```

37. Solution

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

i = 0
Do While i < 99 And usernames(i).ToUpper() <> usr
  i += 1
Loop

If usernames(i).ToUpper() = usr And passwords(i).ToUpper() = pwd Then
```

```
  Console.WriteLine("Login OK!")
Else
  Console.WriteLine("Login Failed!")
End If
```

38. Solution

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

found = False

For i = 0 To 999
  If names(i) = value_str Then
    Console.WriteLine(SSNs(i))
    found = True
  End If
Next

If Not found Then
  value = Int64.Parse(value_str)
  i = 0
  Do While i < 999 And SSNs(i) <> value
    i += 1
  Loop

  If SSNs(i) = value Then
    found = True
    Console.WriteLine(names(i))
  End If
End If

If Not found Then
  Console.WriteLine("This value does not exist")
End If
```

39. Solution

```
Const STUDENTS = 12
Const LESSONS = 6

Sub Main(args As String())
  Dim i, j As Integer
  Dim found, failure As Boolean

  Dim grades(STUDENTS - 1, LESSONS - 1) As Integer
  For i = 0 To STUDENTS - 1
    For j = 0 To LESSONS - 1
      Do
        grades(i, j) = Console.ReadLine()
        failure = False
        If grades(i, j) < 0 Then
          Console.WriteLine("Error! You entered a negative value")
```

```
          failure = True
        ElseIf grades(i, j) > 100 Then
          Console.WriteLine("Error! You entered a value grater than 100")
          failure = True
        End If
      Loop While failure
    Next
  Next

  Dim average(STUDENTS - 1) As Double
  For i = 0 To STUDENTS - 1
    average(i) = 0
    For j = 0 To LESSONS - 1
      average(i) += grades(i, j)
    Next
    average(i) /= LESSONS
  Next

  found = False
  For i = 0 To STUDENTS - 1
    If average(i) < 70 Then
      found = True
      Exit For
    End If
  Next

  If found Then
    Console.WriteLine("There is at least one student that has an average value below 70")
  End If
End Sub
```

40. Solution

```
Sub Main(args As String())
  Dim word, letter As String
  Dim i As Integer

  Dim morseAlphabet As New Dictionary(Of String, String) From {
    {"A", ".-"},
    {"B", "-..."},
    {"C", "-.-."},
    {"D", "-.."},
    {"E", "."},
    {"F", "..-."},
    {"G", "--."},
    {"H", "...."},
    {"I", ".."},
    {"J", ".---"},
    {"K", "-.-"},
    {"L", ".-.."},
    {"M", "--"},
    {"N", "-."},
```

```
    {"O", "---"},
    {"P", ".--."},
    {"Q", "--.-"},
    {"R", ".-."},
    {"S", "..."},
    {"T", "-"},
    {"U", "..-"},
    {"V", "...-"},
    {"W", ".--"},
    {"X", "-..-"},
    {"Y", "-.--"},
    {"Z", "--.."},
    {" ", "/"}
  }

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

  For i = 0 To word.Length - 1
    letter = word(i)
    Console.Write(morseAlphabet(letter.ToUpper()) & " ")
  Next
End Sub
```

Review in “Data Structures in Visual Basic”

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. False
9. True
10. True
11. False
12. True

Chapter 36

36.8 Review Questions: True/False

1. False
2. True
3. False
4. True
5. True
6. False
7. True
8. False
9. True
10. False
11. True
12. True
13. True
14. True
15. True
16. False
17. False
18. True
19. False
20. True
21. True
22. True
23. True
24. False
25. True
26. False
27. True
28. False
29. True
30. True
31. True
32. True
33. False

36.9 Review Exercises

1. Solution

```
Function find_max(a As Integer, b As Integer)
  Dim maximum As Integer
  If a > b Then
    maximum = a
  Else
    maximum = b
  End If
  Return maximum
End Function
```

2. Solution

Step	Statement	Main Code		Method sum_digits()		
		s	i	a	d1	d2
1	s = 0	**0**	?			
2	i = 25	0	**25**			
3	i <= 27	True				
4	s += sum_digits(i)			**25**	?	?
5	d1 = a Mod 10			25	**5**	?
6	d2 = a \ 10			25	5	**2**

7	`Return d1 + d2`	**7**	25			
8	`i += 1`	7	**26**			
9	`i <= 27`	True				
10	`s += sum_digits(i)`			**26**	?	?
11	`d1 = a Mod 10`			26	**6**	?
12	`d2 = a \ 10`			26	6	**2**
13	`Return d1 + d2`	**15**	26			
14	`i += 1`	15	**27**			
15	`i <= 27`	True				
16	`s += sum_digits(i)`			**27**	?	?
17	`d1 = a Mod 10`			27	**7**	?
18	`d2 = a \ 10`			27	7	**2**
19	`Return d1 + d2`	**24**	27			
20	`i += 1`	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	`Do While i < 6`	True				
4	`If i Mod 2 = 1 Then`	True				
5	`s += 1`	**1**	1			
6	`i += 1`	1	**2**			
7	`Do While i < 6`	True				
8	`If i Mod 2 = 1 Then`	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 += 1`			2	1	**2**
15	`k <= a`			True		
16	`total += k`			2	**3**	2
17	`k += 1`			2	3	**3**

18	k <= a			False		
19	Return total	**4**	2			
20	i += 1	4	**3**			
21	Do While i < 6	True				
22	If i Mod 2 = 1 Then	True				
23	s += 1	**5**	3			
24	i += 1	5	**4**			
25	Do While i < 6	True				
26	If i Mod 2 = 1 Then	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 += 1			4	1	**2**
33	k <= a			True		
34	total += k			4	**3**	2
35	k += 1			4	3	**3**
36	k <= a			True		
37	total += k			4	**6**	4
38	k += 1			4	6	**4**
39	k <= a			True		
40	total += k			4	**10**	4
41	k += 1			4	10	**5**
42	k <= a			False		
43	Return total	**15**	4			
44	i += 1	15	**5**			
45	Do While i < 6	True				
46	If i Mod 2 = 1 Then	True				
47	s += 1	**16**	5			
48	i += 1	16	**6**			
49	Do While i < 6	False				
50	Console.WriteLine(s)	It displays: 16				

4. Solution

Step	Statement	Main Code				Method custom_div()	
		k	m	a	x	b	d
1	k = Console.ReadLine()	**12**	?	?	?		
2	m = 2	12	**2**	?	?		
3	a = 1	12	2	**1**	?		
4	Do While a < 6	True					
5	If k Mod m <> 0 Then	False					
6	x = a + m + custom_div(m, a)					2	1
7	Return (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 += 1	12	**3**	3	4		
11	Do While a < 6	True					
12	If k Mod m <> 0 Then	False					
13	x = a + m + custom_div(m, a)					3	3
14	Return (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 += 1	12	**4**	5	9		
18	Do While a < 6	True					
19	If k Mod m <> 0 Then	False					
20	x = a + m + custom_div(m, a)					4	5
21	Return (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 += 1	12	**5**	7	13		
25	Do 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 = Console.ReadLine()	1	**3**	

4	display(x)			**3**
5	If a Mod 2 = 0 Then			False
6	Console.WriteLine(a & " is odd")	It displays: 3 is odd		
7	i += 1	**2**	3	
8	i <= 5	True		
9	x = Console.ReadLine()	2	**7**	
10	display(x)			**7**
11	If a Mod 2 = 0 Then			False
12	Console.WriteLine(a & " is odd")	It displays: 7 is odd		
13	i += 1	**3**	7	
14	i <= 5	True		
15	x = Console.ReadLine()	3	**9**	
16	display(x)			**9**
17	If a Mod 2 = 0 Then			False
18	Console.WriteLine(a & " is odd")	It displays: 9 is odd		
19	i += 1	**4**	9	
20	i <= 5	True		
21	x = Console.ReadLine()	4	**2**	
22	display(x)			**2**
23	If a Mod 2 = 0 Then			True
24	Console.WriteLine(a & " is even")	It displays: 2 is even		
25	i += 1	**5**	2	
26	i <= 5	True		
27	x = Console.ReadLine()	5	**4**	
28	display(x)			**4**
29	If a Mod 2 = 0 Then			True
30	Console.WriteLine(a & " is even")	It displays: 4 is even		
31	i += 1	**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	Do While x Mod y < 30	True			

4	`division(y, x)`			**30**	**20**
5	`b = b \ a`			30	**0**
6	`Console.WriteLine(a * b)`	It displays: 0			
7	`x = 4 * y`	**120**	30		
8	`y += 1`	120	**31**		
9	`Do While x Mod y < 30`	True			
10	`division(y, x)`			**31**	**120**
11	`b = b \ a`			31	**3**
12	`Console.WriteLine(a * b)`	It displays: 93			
13	`x = 4 * y`	**124**	31		
14	`y += 1`	124	**32**		
15	`Do While x Mod y < 30`	True			
16	`division(y, x)`			**32**	**124**
17	`b = b \ a`			32	**3**
18	`Console.WriteLine(a * b)`	It displays: 96			
19	`x = 4 * y`	**128**	32		
20	`y += 1`	128	**33**		
21	`Do While x Mod y < 30`	True			
22	`division(y, x)`			**33**	**128**
23	`b = b \ a`			33	**3**
24	`Console.WriteLine(a * b)`	It displays: 99			
25	`x = 4 * y`	**132**	33		
26	`y += 1`	132	**34**		
27	`Do While x Mod 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 = 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 + j ^ 2`			2	**4**	2

9	`j += 2`			2	4	**4**
10	`j <= 2 * n`				True	
11	`s = s + 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 += 1`	**2**	2			
16	`i <= 3`	True				
17	`m = 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 + j ^ 2`			3	**4**	2
23	`j += 2`			3	4	**4**
24	`j <= 2 * n`				True	
25	`s = s + j ^ 2`			3	**20**	4
26	`j += 2`			3	20	**6**
27	`j <= 2 * n`				True	
28	`s = s + 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 += 1`	**3**	3			
33	`i <= 3`	True				
34	`m = 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 + j ^ 2`			4	**4**	2
40	`j += 2`			4	4	**4**
41	`j <= 2 * n`				True	
42	`s = s + j ^ 2`			4	**20**	4
43	`j += 2`			4	20	**6**
44	`j <= 2 * n`				True	
45	`s = s + j ^ 2`			4	**56**	6

46	`j += 2`			4	56	**8**
47	`j <= 2 * n`				True	
48	`s = s + 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 += 1`	**4**	4			
53	`i <= 3`	False				

8. Solution

```
Function find_sum(a As Double, b As Double, c As Double) As Double
  Return a + b + c
End Function
```

9. Solution

```
Function find_avg(a As Double, b As Double, c As Double, d As Double) As Double
  Return (a + b + c + d) / 4
End Function
```

10. Solution

```
Function maximum(a As Double, b As Double, c As Double) As Double
  Dim m As Double

  m = a
  If b > m Then
    m = b
  End If
  If c > m Then
    m = c
  End If

  Return m
End Function
```

11. Solution

```
Sub display_max(a As Double, b As Double, c As Double, d As Double, e As Double)
  Dim m As Double

  m = a
  If b > m Then
    m = b
  End If

  If c > m Then
    m = c
  End If
```

```
  If d > m Then
    m = d
  End If

  If e > m Then
    m = e
  End If

  Console.WriteLine(m)
End Sub
```

12. Solution

```
Function my_round(x As Double) As Double
  Dim digit_to_check As Integer
  Dim return_value As Double

  digit_to_check = Fix(x * 1000) Mod 10
  If digit_to_check >= 5 Then
    return_value = (Fix(x * 100) + 1) / 100.0
  Else
    return_value = Fix(x * 100) / 100.0
  End If

  Return return_value
End Function
```

13. Solution

```
Function find_min(a As Double, b As Double) As Double
  Dim minimum As Double

  minimum = a
  If b < minimum Then
    minimum = b
  End If
  Return minimum
End Function

Sub Main(args As String())
  Dim temp1, temp2, x1, x2, x3, x4 As Double

  Console.Write("Enter four numbers: ")
  x1 = Console.ReadLine()
  x2 = Console.ReadLine()
  x3 = Console.ReadLine()
  x4 = Console.ReadLine()

  'First approach
  temp1 = find_min(x1, x2)
  temp2 = find_min(x3, x4)
  Console.WriteLine(find_min(temp1, temp2))
```

```
  'Second approach
  Console.WriteLine(find_min(find_min(x1, x2), find_min(x3, x4)))
End Sub
```

14. Solution

```
Function Kelvin_to_Fahrenheit(kelvin As Double) As Double
  Return 1.8 * kelvin - 459.67
End Function

Function Kelvin_to_Celsius(kelvin As Double) As Double
  Return kelvin - 273.15
End Function

Sub Main(args As String())
  Dim k As Double

  Console.Write("Enter a temperature in degrees Kelvin: ")
  k = Console.ReadLine()
  Console.WriteLine("Fahrenheit: " & Kelvin_to_Fahrenheit(k))
  Console.WriteLine("Celsius: " & Kelvin_to_Celsius(k))
End Sub
```

15. Solution

```
Function bmi(w As Double, h As Double) As String
  Dim b As Double
  Dim return_value As String

  b = w * 703 / h ^ 2
  If b < 16 Then
    return_value = "You must add weight."
  ElseIf b < 18.5 Then
    return_value = "You should add some weight."
  ElseIf b < 25 Then
    return_value = "Maintain your weight."
  ElseIf b < 30 Then
    return_value = "You should lose some weight."
  Else
    return_value = "You must lose weight."
  End If

  Return return_value
End Function

Sub Main(args As String())
  Dim height, weight As Double
  Dim age As Integer

  Console.Write("Enter your weight (in pounds): ")
  weight = Console.ReadLine()
  Do While weight < 0
    Console.Write("Error! Enter your weight (in pounds): ")
```

```
    weight = Console.ReadLine()
  Loop

  Console.WriteLine("Enter your age: ")
  age = Console.ReadLine()
  Do While age < 18
    Console.Write("Error! Enter your age: ")
    age = Console.ReadLine()
  Loop

  Console.WriteLine("Enter your height (in inches): ")
  height = Console.ReadLine()
  Do While height < 0
    Console.WriteLine("Error! Enter your height (in inches): ")
    height = Console.ReadLine()
  Loop

  Console.WriteLine(bmi(weight, height))
End Sub
```

16. Solution

```
Sub num_of_days(year As Integer, month As Integer)
  Dim days As Integer

  Select Case month
    Case 4, 6, 9, 11
      days = 30
    Case 2
      If year Mod 4 = 0 And year Mod 100 <> 0 Or year Mod 400 = 0 Then
        days = 29
      Else
        days = 28
      End If
    Case Else
      days = 31
  End Select

  Console.WriteLine(days)
End Sub

Sub Main(args As String())
  Dim m, y As Integer

  Console.Write("Enter a year: ")
  y = Console.ReadLine()
  For m = 1 To 12
    num_of_days(y, m)
  Next
End Sub
```

17. Solution

```
Sub display_menu()
  Console.WriteLine()
  Console.WriteLine("1. Convert meters to miles")
  Console.WriteLine("2. Convert miles to meters")
  Console.WriteLine("3. Exit")
  Console.Write("Enter a choice: ")
End Sub

Sub meters_to_miles(meters As Double)
  Console.WriteLine(meters & " meters equals " & (meters / 1609.344) & " miles")
End Sub

Sub miles_to_meters(miles As Double)
  Console.WriteLine(miles & " miles equals " & (miles * 1609.344) & " meters")
End Sub

Sub Main(args As String())
  Dim choice As Integer
  Dim distance As Double

  display_menu()
  choice = Console.ReadLine()
  Do While choice <> 3
    Console.WriteLine("Enter distance: ")
    distance = Console.ReadLine()
    If choice = 1 Then
      meters_to_miles(distance)
    Else
      miles_to_meters(distance)
    End If

    display_menu()
    choice = Console.ReadLine()
  Loop
End Sub
```

18. Solution

```
Sub amount_to_pay(seconds As Integer)
  Dim extra, tax, total, total_without_tax As Double

  If seconds <= 600 Then
    extra = 0
  ElseIf seconds <= 1200 Then
    extra = (seconds - 600) * 0.01
  Else
    extra = 600 * 0.01 + (seconds - 1200) * 0.02
  End If

  total_without_tax = 10 + extra
  tax = total_without_tax * 11 / 100
```

```
  total = total_without_tax + tax

  Console.WriteLine("Total amount to pay: " & total)
End Sub

Sub Main(args As String())
  Dim seconds As Integer

  Console.Write("Enter number of seconds: ")
  seconds = Console.ReadLine()
  amount_to_pay(seconds)
End Sub
```

Chapter 37

37.10 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. False
19. False
20. False
21. True
22. True
23. False
24. True
25. True
26. True

37.11 Review Exercises

1. Solution

It displays: 5

2. Solution

It displays: 14

3. Solution

It displays: 14

4. Solution

Step	Statement	Global		Main Code		void Method swap()		
		arr(0)	arr(1)	k	x	x	y	temp
1	`k = Console.ReadLine()`	?	?	**12**	?			
2	`arr(1) = 1`	?	**1**	12	?			
3	`arr(0) = 1`	**1**	1	12	?			
4	`Do While arr(0) < 8`	1	1	True				
5	`If k Mod arr(1) <> 0 Then`	1	1	False				
6	`x = arr(0) + arr(1) + Fix(arr(0) - arr(1))`	1	1	12	**2**			
7	`Console.WriteLine(arr(1) & " " & arr(0) & " " & x)`	It displays: 1 1 2						
8	`arr(0) += 2`	**3**	1	12	2			
9	`arr(1) += 1`	3	**2**	12	2			

10	`swap(arr(0), arr(1))`					**3**	**2**	?
11	`temp = x`					3	2	**3**
12	`x = y`					**2**	2	3
13	`y = temp`					2	**3**	3
14	`Do While arr(0) < 8`	**2**	**3**	12	2			
		2	3	True				
15	`If k Mod arr(1) <> 0 Then`	2	3	False				
16	`x = arr(0) + arr(1) + Fix(arr(0) - arr(1))`	2	3	12	**4**			
17	`Console.WriteLine(arr(1) & " " & arr(0) & " " & x)`	It displays: 3 2 4						
18	`arr(0) += 2`	**4**	3	12	4			
19	`arr(1) += 1`	4	**4**	12	4			
20	`swap(arr(0), arr(1))`					**4**	**4**	?
21	`temp = x`					4	4	**4**
22	`x = y`					**4**	4	4
23	`y = temp`					4	**4**	4
24	`Do While arr(0) < 8`	**4**	**4**	12	4			
		4	4	True				
25	`If k Mod arr(1) <> 0 Then`	4	4	False				
26	`x = arr(0) + arr(1) + Fix(arr(0) - arr(1))`	4	4	12	**8**			
27	`Console.WriteLine(arr(1) & " " & arr(0) & " " & x)`	It displays: 4 4 8						
28	`arr(0) += 2`	**6**	4	12	8			
29	`arr(1) += 1`	6	**5**	12	8			
30	`swap(arr(0), arr(1))`					**6**	**5**	?
31	`temp = x`					6	5	**6**
32	`x = y`					**5**	5	6
33	`y = temp`					5	**6**	5
34	`Do While arr(0) < 8`	**5**	**6**	12	8			
		5	6	True				
35	`If k Mod arr(1) <> 0 Then`	5	6	False				
36	`x = arr(0) + arr(1) + Fix(arr(0) - arr(1))`	5	6	12	**10**			
37	`Console.WriteLine(arr(1) & " " & arr(0) & " " & x)`	It displays: 6 5 10						
38	`arr(0) += 2`	**7**	6	12	10			
39	`arr(1) += 1`	7	**7**	12	10			
40	`swap(arr(0), arr(1))`					**7**	**7**	?
41	`temp = x`					7	7	**7**

42	x = y					**7**	7	7
43	y = temp					7	**7**	7
44	Do While arr(0) < 8	**7**	**7**	12	10			
		7	7	True				
45	If k Mod arr(1) <> 0 Then	7	7	True				
46	x = arr(0) Mod arr(1)	7	7	12	**0**			
47	swap(arr(1), arr(0))					**7**	**7**	?
48	temp = x					7	7	**7**
49	x = y					**7**	7	7
50	y = temp					7	**7**	7
51	Console.WriteLine(arr(1) & " " & arr(0) & " " & x)	7	7	12	0			
		It displays: 7 7 0						
52	arr(0) += 2	**9**	7	12	0			
53	arr(1) += 1	9	**8**	12	0			
54	swap(arr(0), arr(1))					**9**	**8**	?
55	temp = x					9	8	**9**
56	x = y					**8**	8	9
57	y = temp					8	**9**	9
58	Do While arr(0) < 8	**8**	**9**	12	0			
		False						

5. Solution

It displays: hellohellohello

6. Solution

It displays: 15

7. Solution

It displays: 11 4

8. Solution

```
Const STUDENTS = 10
Const LESSONS = 5

Sub part1(names() As String, grades(,) As Integer)
  Dim i, j As Integer

  For i = 0 To STUDENTS - 1
    Console.Write("Enter name for student No. " & (i + 1) & ": ")
    names(i) = Console.ReadLine()
    For j = 0 To LESSONS - 1
```

```
      Console.Write("Enter grade for lesson No. " & (j + 1) & ": ")
      grades(i, j) = Console.ReadLine()
    Next
  Next
End Sub

Function part2(grades(,) As Integer) As Double()
  Dim average(STUDENTS - 1) As Double
  Dim i, j As Integer

  For i = 0 To STUDENTS - 1
    average(i) = 0
    For j = 0 To LESSONS - 1
      average(i) += grades(i, j)
    Next
    average(i) /= LESSONS
  Next
  Return average
End Function

Sub part3(average() As Double, names() As String)
  Dim m, n As Integer
  Dim temp As Double
  Dim temp_str As String

  For m = 1 To STUDENTS - 1
    For n = STUDENTS - 1 To m Step -1
      If average(n) > average(n - 1) Then
        temp = average(n)
        average(n) = average(n - 1)
        average(n - 1) = temp

        temp_str = names(n)
        names(n) = names(n - 1)
        names(n - 1) = temp_str
      ElseIf average(n) = average(n - 1) Then
        If names(n).CompareTo(names(n - 1)) < 0 Then
          temp_str = names(n)
          names(n) = names(n - 1)
          names(n - 1) = temp_str
        End If
      End If
    Next
  Next
End Sub

Sub Main(args As String())
  Dim i As Integer

  Dim names(STUDENTS - 1) As String
  Dim grades(STUDENTS - 1, LESSONS - 1) As Integer
  Dim average(STUDENTS - 1) As Double
```

```
  part1(names, grades)

  average = part2(grades)

  part3(average, names)

  For i = 0 To STUDENTS - 1
    Console.WriteLine(names(i) & vbTab & average(i))
  Next
End Sub
```

9. Solution

```
Function part1() As String
  Dim message As String

  Console.Write("Enter a message: ")
  message = Console.ReadLine().ToLower()
  Return message
End Function

Function part2(message As String) As String
  Dim letter, message_clean As String
  Dim i As Integer

  message_clean = ""
  For i = 0 To message.Length - 1
    letter = message(i)
    If letter <> " " And letter <> "," And letter <> "." And letter <> "?" Then
      message_clean += letter
    End If
  Next
  Return message_clean
End Function

Function part3(message_clean As String) As Boolean
  Dim middle_pos, i, j As Integer
  Dim palindrome As Boolean
  Dim left_letter, right_letter As String

  middle_pos = (message_clean.Length - 1) \ 2
  j = message_clean.Length - 1
  palindrome = True
  For i = 0 To middle_pos
    left_letter = message_clean(i)
    right_letter = message_clean(j)
    If left_letter <> right_letter Then
      palindrome = False
      Exit For
    End If
    j -= 1
  Next
  Return palindrome
```

```
End Function

Function part4(message As String) As Boolean
  Dim message_clean As String
  Dim palindrome As Boolean

  message_clean = part2(message)
  palindrome = part3(message_clean)
  Return palindrome
End Function

Sub Main(args As String())
  Dim message As String
  Dim palindrome As Boolean

  message = part1()
  palindrome = part4(message)
  If palindrome Then
    Console.WriteLine("The message is palindrome")
  End If
End Sub
```

10. Solution

```
Sub Main(args As String())
  Dim a, b, c, maximum As Integer

  a = Console.ReadLine()
  b = Console.ReadLine()
  c = Console.ReadLine()
  d = Console.ReadLine()

  maximum = a
  If b > maximum Then
    maximum = b
  End If
  If c > maximum Then
    maximum = c
  End If
  If d > maximum Then
    maximum = d
  End If

  Console.WriteLine(maximum)
End Sub
```

11. Solution

```
Sub f1(a As Double, b As Double, c As Double, returning_array() As Double)
  returning_array(0) = a + b + c
  returning_array(1) = returning_array(0) / 3
End Sub
```

12. Solution

```
Function my_round(x As Double, Optional decimal_places As Integer = 2) As Double
  Dim return_value As Double

  Dim digit_to_check As Integer = Fix(x * 10 ^ (decimal_places + 1)) Mod 10
  If digit_to_check >= 5 Then
    return_value = Fix(x * 10 ^ decimal_places + 1) / 10 ^ decimal_places
  Else
    return_value = Fix(x * 10 ^ decimal_places) / 10 ^ decimal_places
  End If
  Return return_value
End Function
```

13. Solution

```
Function get_input() As Boolean
  Dim answer As String

  Do
    Console.Write("Enter Yes or No: ")
    answer = Console.ReadLine().ToUpper()
  Loop While answer <> "YES" And answer <> "NO"

  Return answer = "YES"  'This returns True or False
End Function

Function find_area(b As Double, h As Double) As Double
  Return b * h
End Function

Sub Main(arghs As String())
  Dim b, height As Double

  Do
    Console.Write("Enter the base of the parallelogram: ")
    b = Console.ReadLine()
    Console.Write("Enter the height of the parallelogram: ")
    h = Console.ReadLine()

    Console.WriteLine("Area = " & find_area(b, h))

    Console.WriteLine("Would you like to repeat? ")
  Loop While get_input()
End Sub
```

14. Solution

```
Const STUDENTS = 100

Sub get_arrays(names() As String, grades() As Integer)
  Dim i As Integer

  For i = 0 To STUDENTS - 1
```

```
    Console.Write("Enter name: ")
    names(i) = Console.ReadLine()
    Console.Write("Enter grade: ")
    grades(i) = Console.ReadLine()
  Next
End Sub

Function get_average(grades() As Integer) As Double
  Dim i, total As Integer = 0
  For i = 0 To STUDENTS - 1
    total += grades(i)
  Next
  Return total / STUDENTS
End Function

Sub sort_arrays(grades() As Integer, names() As String)
  Dim m, n, element_grds As Integer
  Dim element_nms As String

  For m = 1 To STUDENTS - 1
    element_grds = grades(m)
    element_nms = names(m)

    n = m
    Do While n > 0
      If element_grds > grades(n - 1) Then Exit Do
      grades(n) = grades(n - 1)
      names(n) = names(n - 1)
      n -= 1
    Loop

    grades(n) = element_grds
    names(n) = element_nms
  Next
End Sub

Sub Main(args As String())
  Dim i As Integer
  Dim average As Double

  Dim names(STUDENTS - 1) As String
  Dim grades(STUDENTS - 1) As Integer

  get_arrays(names, grades)
  average = get_average(grades)
  sort_arrays(grades, names)
  For i = 0 To STUDENTS - 1
    If grades(i) < average Then
      Console.WriteLine(names(i))
    End If
  Next
End Sub
```

15. Solution

```
Const JUDGES = 10

Function get_array() As Integer()
  Dim score(JUDGES - 1) As Integer
  Dim i As Integer

  For i = 0 To JUDGES - 1
    Console.Write("Judge No " & (i + 1) & ". Enter score: ")
    score(i) = Console.ReadLine()
  Next
  Return score
End Function

Sub find_min_max(score() As Integer, ByRef minimum As Integer, ByRef maximum As Integer)
  Dim i As Integer
  minimum = score(0)
  maximum = score(0)
  For i = 1 To JUDGES - 1
    If score(i) > maximum Then
      maximum = score(i)
    End If
    If score(i) < minimum Then
      minimum = score(i)
    End If
  Next
End Sub

Sub Main(args As String())
  Dim name As String
  Dim total, i, points, minimum, maximum As Integer

  Console.Write("Enter artist's name: ")
  name = Console.ReadLine()
  Dim score() As Integer = get_array()
  find_min_max(score, minimum, maximum)

  total = 0
  For i = 0 To JUDGES - 1
    total += score(i)
  Next

  points = total - minimum - maximum
  Console.WriteLine("Artist " & name & " got " & points & " points")
End Sub
```

16. Solution

```
Function woc(index As Integer) As Double
  Dim return_value As Double

  If index = 1 Then
```

```
    return_value = 1
  Else
    return_value = 2 * woc(index - 1)
  End If
  Return return_value
End Function

Sub Main(args As String())
  Dim total As Double
  Dim i As Integer

  total = 0
  For i = 1 To 64
    total += woc(i)
  Next
  Console.WriteLine(total)
End Sub
```

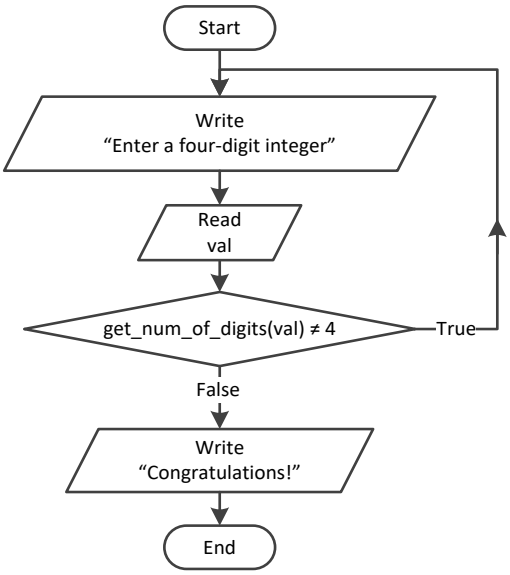
17. Solution

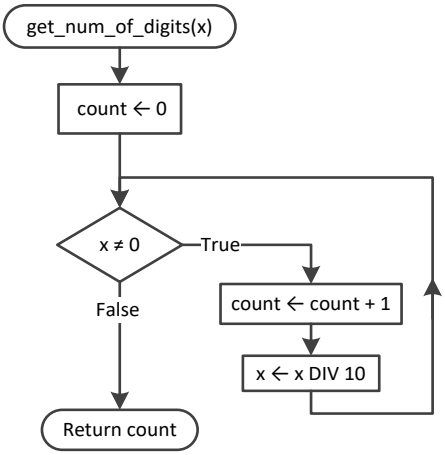
```
Function factorial(value As Integer) As Double
  Dim return_value As Double

  If value = 1 Then
    return_value = 1
  Else
    return_value = value * factorial(value - 1)
  End If

  Return return_value
End Function

Function my_cos(x As Double, Optional i As Integer = 40) As Double
  Dim return_value As Double

  If i = 0 Then
    return_value = 1
  Else
    return_value = my_cos(x, i - 4) + x ^ i / factorial(i) - x ^ (i - 2) / factorial(i - 2)
  End If

  Return return_value
End Function

Sub Main(args As String())
  Console.WriteLine(my_cos(Math.PI / 4))
End Sub
```

Chapter 38

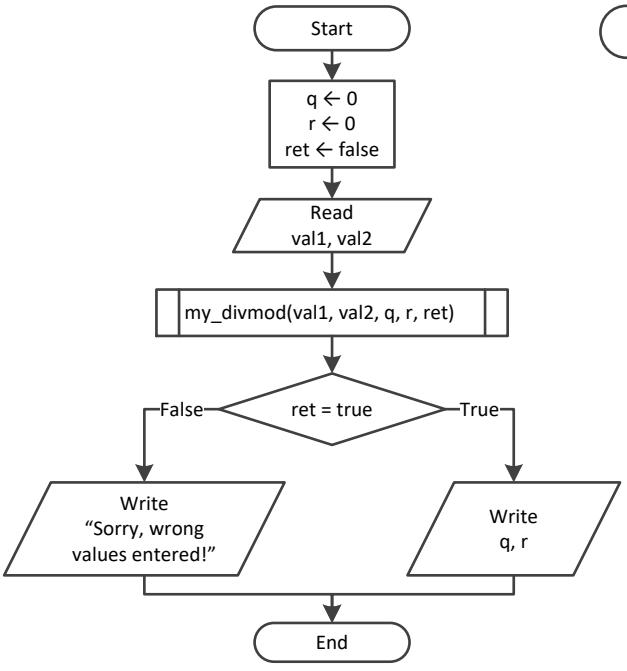
38.3 Review Exercises

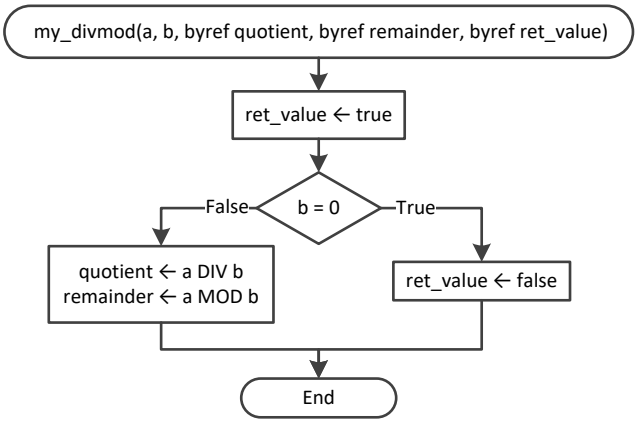
1. Solution





2. Solution

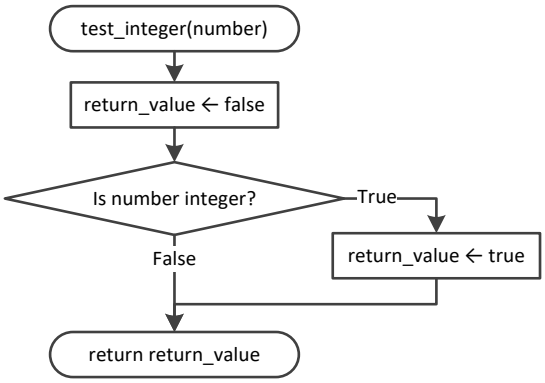


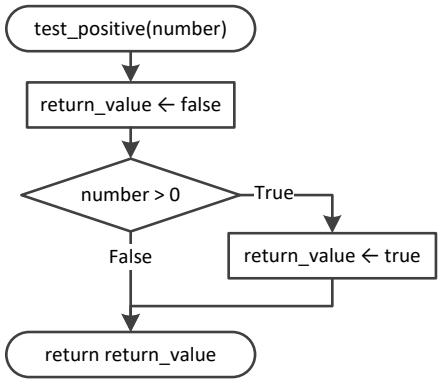


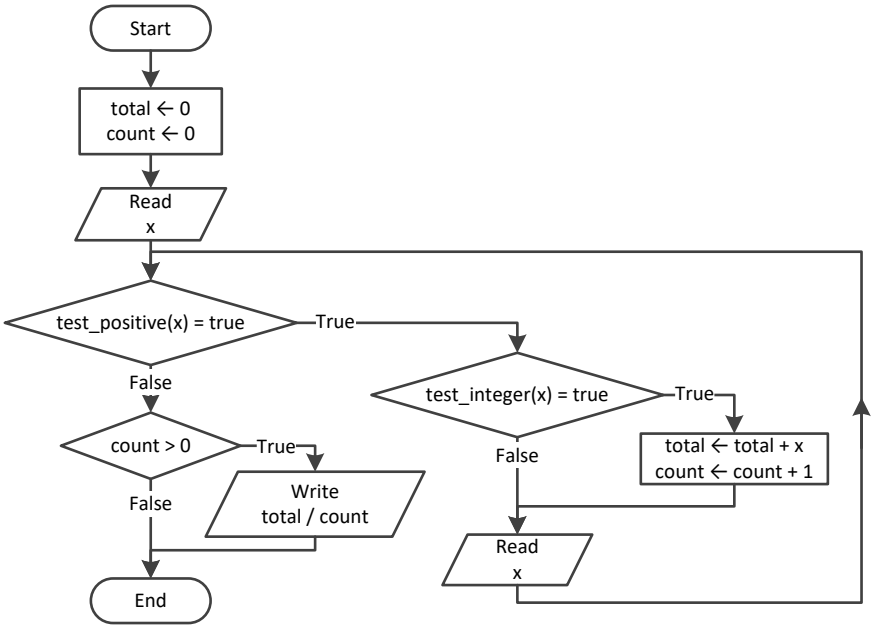
✎ *Flowcharts are a loose method of representing an algorithm. Thus, you can represent a pass by reference using the keyword* `byref`*, which clearly denotes what it actually does.*

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

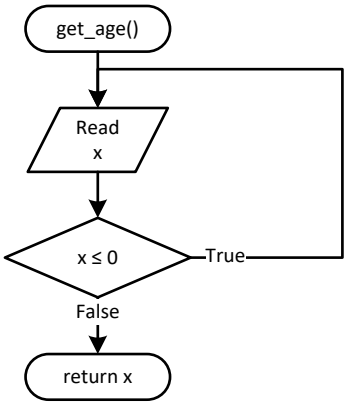
3. Solution

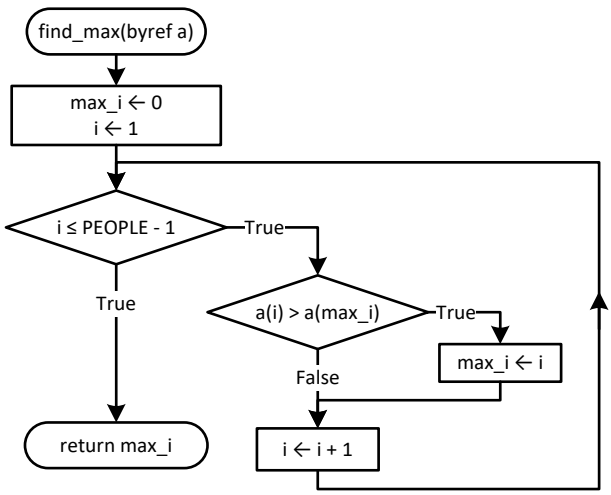


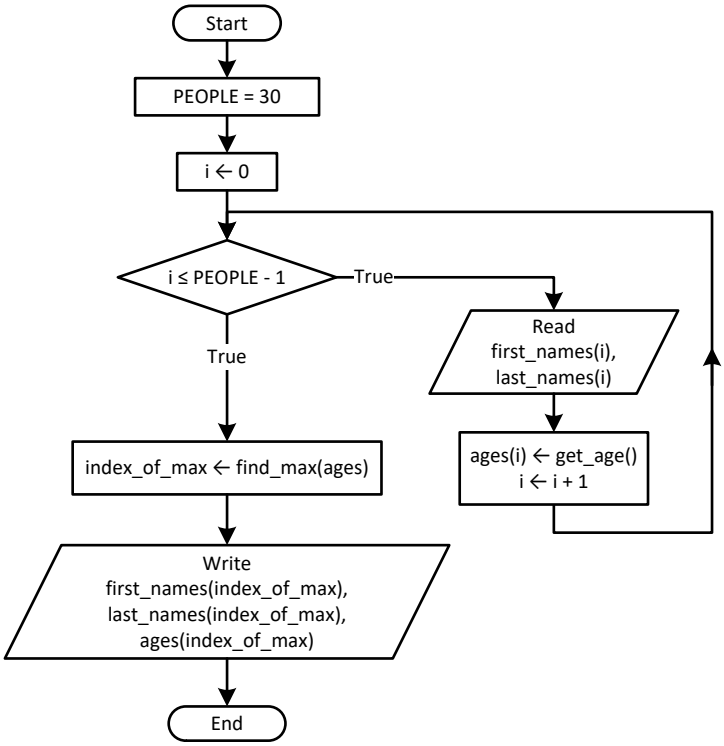




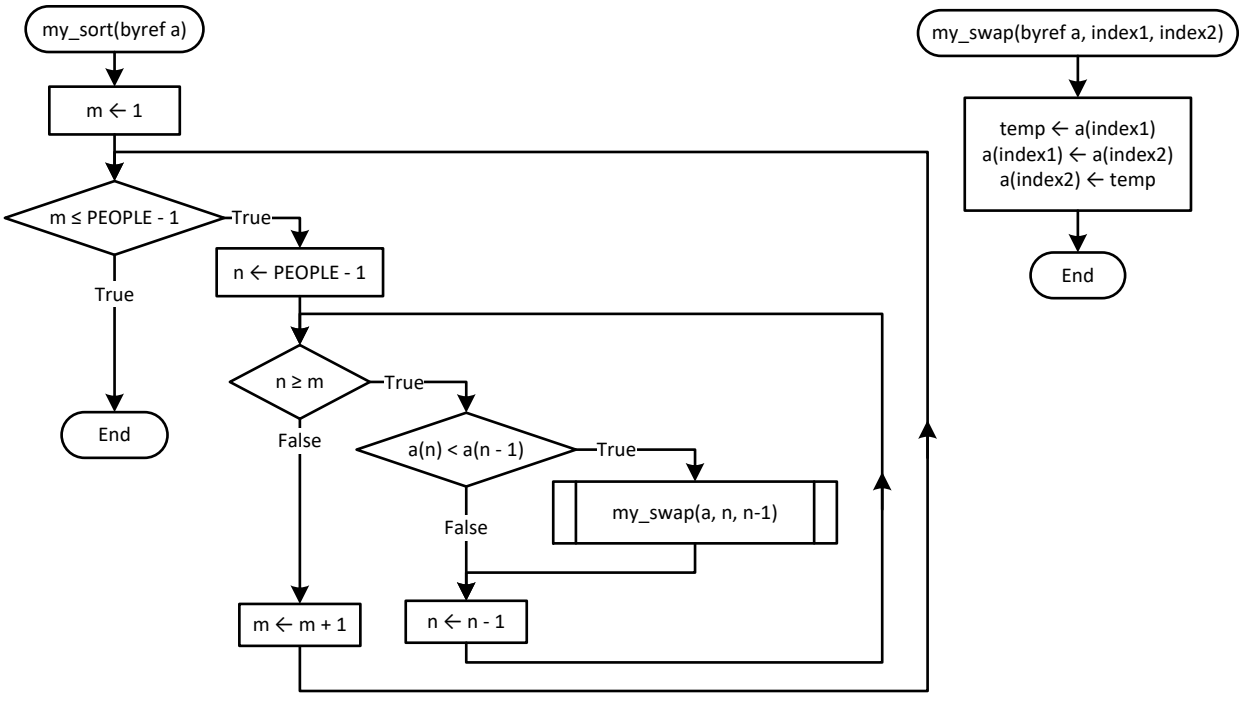
4. Solution

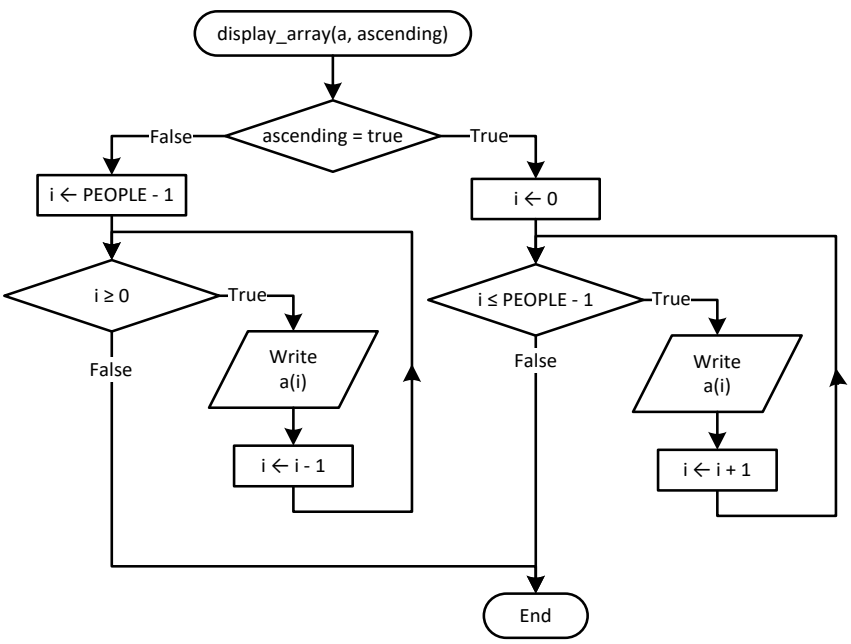


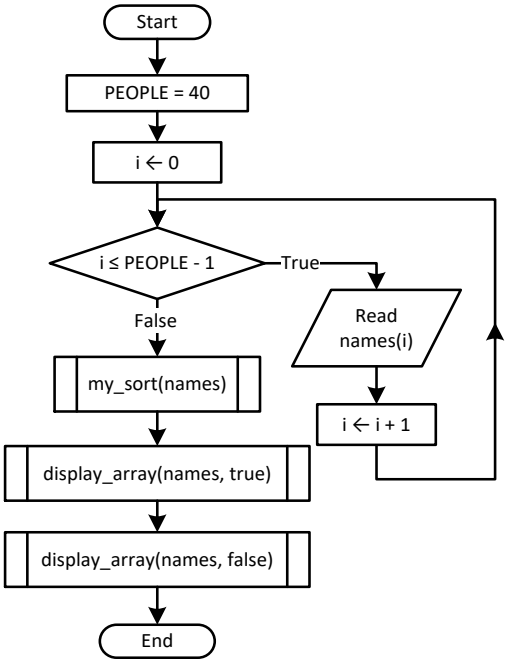



Start
PEOPLE = 30
i ← 0
i ≤ PEOPLE - 1
True
True
Read
first_names(i),
last_names(i)
ages(i) ← get_age()
i ← i + 1
index_of_max ← find_max(ages)
Write
first_names(index_of_max),
last_names(index_of_max),
ages(index_of_max)
End

5. Solution


my_sort(byref a)
m ← 1
m ≤ PEOPLE - 1
True
True
n ← PEOPLE - 1
End
n ≥ m
True
False
a(n) < a(n - 1)
True
False
my_swap(a, n, n-1)
m ← m + 1
n ← n - 1
my_swap(byref a, index1, index2)
temp ← a(index1)
a(index1) ← a(index2)
a(index2) ← temp
End





6. Solution

```
Function f1(n As Integer) As Double
  Dim s As Double
  Dim i As Integer

  s = 0
  For i = 1 To n
    If i < n / 2 Then
      s += n ^ 2
    Else
      s += n ^ 3
    End If
  Next
  Return s
End Function

Sub Main(args As String())
  Dim val As Integer

  Do
    Console.Write("Enter a positive integer ")
    val = Console.ReadLine()
  Loop While val < 0
  Console.WriteLine(f1(val))
End Sub
```

7. Solution

```
Const ELEMENTS = 100

Function read_values() As Double()
```

```
  Dim values(ELEMENTS - 1) As Double
  Dim i As Integer

  For i = 0 To ELEMENTS - 1
    values(i) = Console.ReadLine()
  Next
  Return values
End Function

Sub find_min_max(values() As Double, ByRef min_i As Integer, ByRef max_i As Integer)
  Dim i As Integer

  min_i = 0
  max_i = 0
  For i = 1 To ELEMENTS - 1
    If values(i) < values(min_i) Then
      min_i = i
    End If
    If values(i) > values(max_i) Then
      max_i = i
    End If
  Next
End Sub

Sub Main(args As String())
  Dim v(ELEMENTS - 1) As Double
  Dim min_i, max_i As Integer

  v = read_values()
  find_min_max(v, min_i, max_i)
  Console.WriteLine(v(min_i) & ", " & v(max_i))
End Sub
```

✎ *Please note the way the void method* `find_min_max()` *finds the index positions of the minimum and the maximum values of the array* `values`*. This method is not the same as the one you learned in paragraph 34.3; however, it can be used as an alternative.*

8. Solution

```
Const ACCURACY = 0.000000001

Function factorial(n As Integer) As Double
  Dim i As Integer

  Dim return_value As Double = 1
  For i = 1 To n
    return_value *= i
  Next
  Return return_value
End Function

Function my_sin(x As Double) As Double
  Dim i, sign As Integer
```

```
  Dim sinus, sinus_previous As Double
  sign = 1
  sinus = 0
  i = 1
  Do
    sinus_previous = sinus
    sinus += sign * x ^ i / factorial(i)

    sign = -sign
    i += 2
  Loop While Math.Abs(sinus - sinus_previous) > ACCURACY
  Return sinus
End Function

Function degrees_to_rad(degrees As Double) As Double
  Return 2 * Math.PI * degrees / 360
End Function

Sub Main(args As String())
  Dim i As Integer

  For i = 0 To 360
    Console.WriteLine("sin(" & i & ") ~= " & my_sin(degrees_to_rad(i)))
  Next
End Sub
```

9. Solution

```
Function is_leap(year As Integer) As Boolean
  Dim return_value As Boolean = False
  If year Mod 4 = 0 And year Mod 100 <> 0 Or year Mod 400 = 0 Then
    return_value = True
  End If
  Return return_value
End Function

Function num_of_days(year As Integer, month As Integer) As Integer
  Dim days As Integer

  Select Case month
    Case 4, 6, 9, 11
      days = 30
    Case 2
      If is_leap(year) Then
        days = 29
      Else
        days = 28
      End If
    Case Else
      days = 31
    End Select

  Return days
```

```
End Function

Function check_date(day As Integer, month As Integer, year As Integer) As Boolean
  Dim return_value As Boolean = True
  If month < 1 Or month > 12 Then
    return_value = False
  ElseIf day < 1 Or day > num_of_days(year, month) Then
    return_value = False
  End If
  Return return_value
End Function

Sub Main(args As String())
  Dim day, month, year, total, i As Integer

  Console.Write("Enter day: ")
  day = Console.ReadLine()
  Console.Write("Enter month: ")
  month = Console.ReadLine()
  Console.Write("Enter year: ")
  year = Console.ReadLine()
  Do While Not check_date(day, month, year)
    Console.WriteLine("Error!")
    Console.Write("Enter day: ")
    day = Console.ReadLine()
    Console.Write("Enter month: ")
    month = Console.ReadLine()
    Console.Write("Enter year: ")
    year = Console.ReadLine()
  Loop

  total = 0
  For i = 1 To month - 1
    total += num_of_days(year, i)
  Next
  total += day

  Console.WriteLine(total)
End Sub
```

10. Solution

```
Sub display_menu()
  Console.WriteLine("---------------------------------------------")
  Console.WriteLine("1. Convert USD to Euro (EUR)")
  Console.WriteLine("2. Convert USD to British Pound Sterling (GBP)")
  Console.WriteLine("3. Convert EUR to USD")
  Console.WriteLine("4. Convert EUR to GBP")
  Console.WriteLine("5. Convert GBP to USD")
  Console.WriteLine("6. Convert GBP to EUR")
  Console.WriteLine("7. Exit")
  Console.WriteLine("---------------------------------------------")
```

```
  Console.Write("Enter a choice: ")
End Sub

Function USD_to_EUR(value As Double) As Double
  Return value * 0.87
End Function

Function USD_to_GBP(value As Double) As Double
  Return value * 0.76
End Function

Sub Main(args As String())
  Dim choice As Integer
  Dim amount As Double

  display_menu()
  choice = Console.ReadLine()
  Do While choice <> 7
    Console.Write("Enter an amount: ")
    amount = Console.ReadLine()
    Select Case choice
      Case 1
        Console.WriteLine(amount & " USD = " & USD_to_EUR(amount) & " Euro")
      Case 2
        Console.WriteLine(amount & " USD = " & USD_to_GBP(amount) & " GBP")
      Case 3
        Console.WriteLine(amount & " EUR = " & 1 / USD_to_EUR(1 / amount) & " USD")
      Case 4
        Console.WriteLine(amount & " EUR = " & USD_to_GBP(1 / USD_to_EUR(1 / amount)) & " GBP")
      Case 5
        Console.WriteLine(amount & " GBP = " & 1 / USD_to_GBP(1 / amount) & " USD")
      Case 6
        Console.WriteLine(amount & " GBP = " & USD_to_EUR(1 / USD_to_GBP(1 / amount)) & " EUR")
    End Select

    display_menu()
    choice = Console.ReadLine()
  Loop
End Sub
```

11. Solution

```
Function dice() As Integer
  Dim rnd As New Random()
  Return rnd.Next(1, 7)
End Function

Sub Main(args As String())
  Dim dice1, dice2, i, player, total, total_player1, total_player2
  Dim names(1) As String

  Console.Write("Player1 - Enter name: ")
  names(0) = Console.ReadLine()
```

```
  Console.Write("Player2 - Enter name: ")
  names(1) = Console.ReadLine()

  For player = 0 To 1
    total = 0
    For i = 1 To 10
      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
    Next
    If player = 1 Then
      total_player1 = total
    Else
      total_player2 = total
    End If
  Next

  If total_player1 = total_player2 Then
    Console.WriteLine("Tie!")
  ElseIf total_player1 > total_player2 Then
    Console.WriteLine(names(0) & " wins")
  Else
    Console.WriteLine(names(1) & " wins")
  End If
End Sub
```

12. Solution

```
Const GAS = 1
Const DIESEL = 2
Const HYBRID = 3
Const TAX_RATE = 0.10
Const CARS = 40

Function get_choice() As Integer
  Console.WriteLine("1. Gas")
  Console.WriteLine("2. Diesel")
  Console.WriteLine("3. Hybrid")
  Console.Write("Enter type of the car: ")
  Return Console.ReadLine()
End Function

Function get_days() As Integer
  Console.Write("Enter total number of rental days: ")
  Return Console.ReadLine()
End Function

Function get_charge(car_type As Integer, rental_days As Integer) As Double
  Dim charge As Double
```

```
  If car_type = GAS Then
    If rental_days <= 5 Then
      charge = rental_days * 24
    ElseIf rental_days <= 8 Then
      charge = 5 * 24 + (rental_days - 5) * 22
    Else
      charge = 5 * 24 + 3 * 22 + (rental_days - 8) * 18
    End If
  ElseIf car_type = DIESEL Then
    If rental_days <= 5 Then
      charge = rental_days * 28
    ElseIf rental_days <= 8 Then
      charge = 5 * 28 + (rental_days - 5) * 25
    Else
      charge = 5 * 28 + 3 * 25 + (rental_days - 8) * 21
    End If
  Else
    If rental_days <= 5 Then
      charge = rental_days * 30
    ElseIf rental_days <= 8 Then
      charge = 5 * 30 + (rental_days - 5) * 28
    Else
      charge = 5 * 30 + 3 * 28 + (rental_days - 8) * 23
    End If
  End If
  charge = charge * (1 + TAX_RATE) 'This is equivalent to charge += charge * TAX_RATE
  Return charge
End Function

Sub Main(args As String())
  Dim count, i As Integer
  Dim charge, total As Double

  Dim rented_car_types(CARS - 1) As Integer
  Dim rented_days(CARS - 1) As Integer

  For i = 0 To CARS - 1
    rented_car_types(i) = get_choice()
    rented_days(i) = get_days()
  Next

  total = 0
  For i = 0 To CARS - 1
    charge = get_charge(rented_car_types(i), rented_days(i))
    Console.WriteLine("Car No " & (i + 1) & ": " & charge)
    total += charge
  Next

  count = 0
  For i = 0 To CARS - 1
    If rented_car_types(i) = HYBRID Then
```

```
      count += 1
    End If
  Next

  Console.WriteLine("Hybrids rented: " & count)
  Console.WriteLine("Net profit: " & total / (1 + TAX_RATE))
End Sub
```

13. Solution

```
Const CHANNELS = 10
Const DAYS = 7

Sub get_data(names() As String, viewers(,) As Integer)
  Dim i, j As Integer
  Dim day_names() As String = {"Monday", "Tuesday", "Wednesday",
                               "Thursday", "Friday", "Saturday", "Sunday"}

  For i = 0 To CHANNELS - 1
    Console.WriteLine("Enter name for channel No. " & (i + 1) & ": ")
    names(i) = Console.ReadLine()
    For j = 0 To DAYS - 1
      Console.Write("Enter the number of viewers of the main news program on " & day_names(j))
      Console.Write(" for channel " & names(i) & ": ")
      viewers(i, j) = Console.ReadLine()
    Next
  Next
End Sub

Function get_average(a() As Integer) As Double
  Dim total ,i As Integer

  total = 0
  For i = 0 To 4
    total += a(i)
  Next
  Return total / 5.0
End Function

Sub Main(args As String())
  Dim i, j As Integer
  Dim weekend As Double
  Dim increasing As Boolean

  Dim names(CHANNELS - 1) As String
  Dim viewers(CHANNELS - 1, DAYS - 1) As Integer
  get_data(names, viewers)

  Dim temporary_array(4) As Integer
  For i = 0 To CHANNELS - 1
    For j = 0 To 4
      temporary_array(j) = viewers(i, j)
    Next
```

```
    weekend = (viewers(i, DAYS - 2) + viewers(i, DAYS - 1)) / 2
    If weekend >= 1.2 * get_average(temporary_array) Then
      Console.WriteLine(names(i))
    End If
  Next

  For i = 0 To CHANNELS - 1
    increasing = True
    For j = 1 To DAYS - 1
      If viewers(i, j) <= viewers(i, j - 1) Then
        increasing = False
      End If
    Next
    If increasing Then
      Console.WriteLine(names(i))
    End If
  Next
End Sub
```

14. Solution

```
Const CITIZENS = 300

Sub input_data(SSNs() As Long, answers() As String)
  Dim i As Integer

  For i = 0 To CITIZENS - 1
    Console.Write("Enter SSN: ")
    SSNs(i) = Int64.Parse(Console.ReadLine())
    Console.Write("Enter answer: ")
    answers(i) = Console.ReadLine()
  Next
End Sub

Sub sort_arrays(SSNs() As Long, answers() As String)
  Dim m, n, index_of_min As Integer
  Dim minimum, temp As Long
  Dim temp_str As String

  For m = 0 To CITIZENS - 1
    minimum = SSNs(m)
    index_of_min = m
    For n = m To CITIZENS - 1
      If SSNs(n) < minimum Then
        minimum = SSNs(n)
        index_of_min = n
      End If
    Next
    temp = SSNs(m)
    SSNs(m) = SSNs(index_of_min)
    SSNs(index_of_min) = temp
    temp_str = answers(m)
```

```
    answers(m) = answers(index_of_min)
    answers(index_of_min) = temp_str
  Next
End Sub

Function search_array(SSNs() As Long, SSN As Long) As Integer
  Dim left, right, middle, index_position, return_value
  Dim found As Boolean

  left = 0
  right = CITIZENS - 1
  found = False
  Do While left <= right And Not found
    middle = (left + right) \ 2

    If SSNs(middle) > SSN Then
      right = middle - 1
    ElseIf SSNs(middle) < SSN Then
      left = middle + 1
    Else
      found = True
      index_position = middle
    End If
  Loop

  If Not found Then
    Console.WriteLine("SSN not found!")
    return_value = -1
  Else
    return_value = index_position
  End If
  Return return_value
End Function

Function count_answers(answers() As String, answer As String) As Integer
  Dim count, i As Integer

  count = 0
  For i = 0 To CITIZENS - 1
    If answers(i) = answer Then
      count += 1
    End If
  Next
  Return count
End Function

Sub Main(args As String())
  Dim SSNs(CITIZENS - 1) As Long
  Dim SSN As Long
  Dim answers(CITIZENS - 1) As String
  Dim index, count As Integer
  Dim answer As String
```

```
  Do
    input_data(SSNs, answers)
    sort_arrays(SSNs, answers)

    Console.Write("Enter an SSN to search: ")
    SSN = Int64.Parse(Console.ReadLine())

    index = search_array(SSNs, SSN)
    If index <> -1 Then
      answer = answers(index)
      Console.WriteLine(answer)

      count = count_answers(answers, answer)
      Console.WriteLine(count * 100 / CITIZENS)
    End If
    Console.WriteLine("Repeat? ")
    answer = Console.ReadLine()
  Loop While answer = "yes"
End Sub
```

15. Solution

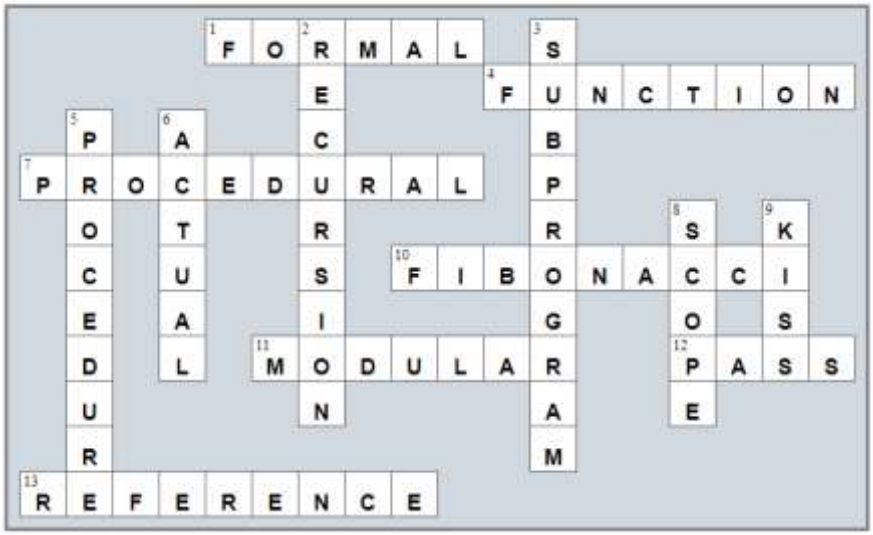
```
Const TEAMS = 8
Const GAMES = 12

Sub input_data(names() As String, results(,) As String)
  Dim i, j As Integer

  For i = 0 To TEAMS - 1
    Console.Write("Enter team name: ")
    names(i) = Console.ReadLine()
    For j = 0 To GAMES - 1
      Console.Write("Enter result (W, L, T): ")
      results(i, j) = Console.ReadLine()
    Next
  Next
End Sub

Sub display_result(names() As String, results(,) As String)
  Dim result As String
  Dim i, j As Integer
  Dim found As Boolean

  Console.Write("Enter a result to search (W, L, T): ")
  result = Console.ReadLine()
  For i = 0 To TEAMS - 1
    Console.WriteLine("Team: " & names(i))
    found = False
    For j = 0 To GAMES - 1
      If results(i, j) = result Then
        Console.WriteLine("Week: " & (j + 1))
        found = True
      End If
```

```
    Next
    If Not found Then
      Console.WriteLine("Nothing found")
    End If
  Next
End Sub

Function find_team(names() As String) As Integer
  Dim name As String
  Dim i, return_value As Integer

  Console.Write("Enter a name to search: ")
  name = Console.ReadLine()

  i = 0
  Do While i < TEAMS - 1 And names(i) <> name
    i += 1
  Loop

  If names(i) <> name Then
    return_value = -1
  Else
    return_value = i
  End If
  Return return_value
End Function

Sub Main(args As String())
  Dim names(TEAMS - 1) As String
  Dim results(TEAMS - 1, GAMES - 1) As String
  Dim j, index, total As Integer

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

  index = find_team(names)
  Do While index <> -1
    total = 0
    For j = 0 To GAMES - 1
      If results(index, j) = "W" Then
        total += 3
      ElseIf results(index, j) = "T" Then
        total += 1
      End If
    Next
    Console.WriteLine("Points: " & total)
    index = find_team(names)
  Loop

  If index = -1 Then
    Console.WriteLine("Team not found")
  End If
End Sub
```

Review in “Subprograms”

Review Crossword Puzzle

1.



Chapter 39

39.9 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

39.10 Review Exercises

1. Solution

```
Class Trigonometry
  Public Function square_area(side As Double) As Double
    Return side ^ 2
  End Function

  Public Function rectangle_area(b As Double, h As Double) As Double
    Return b * h
  End Function

  Public Function triangle_area(b As Double, h As Double) As Double
    Return b * h / 2
  End Function
End Class

Sub Main(args As String())
  Dim sqr_side, rctngl_base, rctngl_height, trngl_base, trngl_height As Double
  Dim tr As New Trigonometry()

  Console.Write("Enter square side: ")
  sqr_side = Console.ReadLine()

  Console.Write("Enter rectangle base:  ")
  rctngl_base = Console.ReadLine()
  Console.Write("Enter rectangle height: ")
  rctngl_height = Console.ReadLine()

  Console.Write("Enter triangle base: ")
  trngl_base = Console.ReadLine()
  Console.Write("Enter triangle height: ")
  trngl_height = Console.ReadLine()

  Console.WriteLine(tr.square_area(sqr_side))
  Console.WriteLine(tr.rectangle_area(rctngl_base, rctngl_height))
  Console.WriteLine(tr.triangle_area(trngl_base, trngl_height))
End Sub
```

2. Solution

```
Class Pet
  Public kind As String
  Public legs_number As Integer

  Public Sub start_running()
    Console.WriteLine("Pet is running")
  End Sub

  Public Sub stop_running()
    Console.WriteLine("Pet stopped")
  End Sub
End Class

Sub Main(args As String())
  Dim pet1 As New Pet()
  pet1.kind = "dog"
  pet1.legs_number = 4

  Dim pet2 As New Pet()
  pet2.kind = "monkey"
  pet2.legs_number = 2

  pet1.start_running()
  pet2.start_running()
  pet1.stop_running()
End Sub
```

3. Solution

```
Class Pet
  Private _kind As String
  Private _legs_number As Integer

  'Define the constructor
  Public Sub New(kind As String, legs_number As Integer)
    Me.Kind = kind
    Me.Legs_number = legs_number
  End Sub

  'Define public property Kind
  Public Property Kind
    Get
      Return Me._kind
    End Get

    Set
      If Value <> "" Then
        Me._kind = Value
      Else
        Throw New Exception("Cannot be empty")
      End If
```

```
    End Set
  End Property

  'Define public property Legs_number
  Public Property Legs_number
    Get
      Return Me._legs_number
    End Get

    Set
      If Value >= 0 Then
        Me._legs_number = Value
      Else
        Throw New Exception("Cannot be negative")
      End If
    End Set
  End Property

  Public Sub start_running()
    Console.WriteLine("Pet is running")
  End Sub

  Public Sub stop_running()
    Console.WriteLine("Pet stopped")
  End Sub
End Class

Sub Main(args As String())
  Dim pet1 As New Pet("dog", 4)

  pet1.start_running()
  pet1.stop_running()

  pet1.Kind = "" 'This will throw an error
  pet1.Legs_number = -1 'This will throw an error
End Sub
```

4. Solution

```
Const BOXES = 3

Class Box
  Private _width As Double
  Private _length As Double
  Private _height As Double

  'Define the constructor
  Public Sub New (w As Double, l As Double, h As Double)
    'Initialize fields
    Me._width = w
    Me._length = l
    Me._height = h
  End SUb
```

```
  Public Sub display_volume()
    Console.WriteLine("Volume: " & (Me._width * Me._length * Me._height))
  End Sub

  Public Sub display_dimensions()
    Console.WriteLine(Me._width & " x " & Me._length & " x " & Me._height)
  End Sub
End Class

Sub Main(args As String())
  Dim i As Integer
  Dim w, l, h As Double

  Dim list_of_obj(BOXES) As Box  'Create an array

  For i = 0 To BOXES - 1
    Console.Write("Enter width: ")
      w = Console.ReadLine()
      Console.Write("Enter length: ")
      l = Console.ReadLine()
      Console.Write("Enter height: ")
      h = Console.ReadLine()

      'Add each new object to the array
      list_of_obj(i) = New Box(w, l, h)
  Next

  For i = 0 To BOXES - 1
      list_of_obj(i).display_dimensions()
      list_of_obj(i).display_volume()
  Next
End Sub
```

5. Solution

```
Const BOXES = 3

Class Box
  Private _width As Double
  Private _length As Double
  Private _height As Double

  'Define the constructor
  Public Sub New(w As Double, l As Double, h As Double)
    'Initialize fields (using the corresponding properties)
    Me.Width = w
    Me.Length = l
    Me.Height = h
  End Sub

  'Define public property Width
  Public Property Width
    'Define the getter
```

```
  Get
    Return Me._width
  End Get

  'Define the setter
  Set
    If Value > 0 Then
      Me._width = Value
    Else
      Throw New Exception("Cannot be negative or zero")
    End If
  End Set
End Property

'Define public property Length
Public Property Length
  'Define the getter
  Get
    Return Me._length
  End Get

  'Define the setter
  Set
    If Value > 0 Then
      Me._length = Value
    Else
      Throw New Exception("Cannot be negative or zero")
    End If
  End Set
End Property

'Define public property Height
Public Property Height
  'Define the getter
  Get
    Return Me._height
  End Get

  'Define the setter
  Set
    If Value > 0 Then
      Me._height = Value
    Else
      Throw New Exception("Cannot be negative or zero")
    End If
  End Set
End Property

Public Sub display_volume()
  Console.WriteLine("Volume: " & (Me.Width * Me.Length * Me.Height))
End Sub

Public Sub display_dimensions()
```

```
    Console.WriteLine(Me.Width & " x " & Me.Length & " x " & Me.Height)
  End Sub
End Class

Sub Main(args As String())
  Dim i As Integer
  Dim w, l, h As Double

  Dim list_of_obj(BOXES) As Box 'Create an array

  For i = 0 To BOXES - 1
    Console.Write("Enter width: ")
    w = Console.ReadLine()
    Console.Write("Enter length: ")
    l = Console.ReadLine()
    Console.Write("Enter height: ")
    h = Console.ReadLine()

    'Add each new object to the array
    list_of_obj(i) = New Box(w, l, h)
  Next

  For i = 0 To BOXES - 1
    list_of_obj(i).display_dimensions()
    list_of_obj(i).display_volume()
  Next
End Sub
```

6. Solution

```
Class Cube
  Private _edge As Double

  'Define the constructor
  Public Sub New(edge As Double)
    Me._edge = edge
  End Sub

  Public Sub display_volume()
    Console.WriteLine("Volume: " & Me._edge ^ 3)
  End Sub

  Public Sub display_one_surface()
    Console.WriteLine("One surface: " & Me._edge ^ 2)
  End Sub

  Public Sub display_total_surface()
    Console.WriteLine("Total surface: " & 6 * Me._edge ^ 2)
  End Sub
End Class

Sub Main(args As String())
  Dim edge As Double
```

```
  Console.Write("Enter edge length of a cube: ")
  edge = Console.ReadLine()

  Dim cube1 As New Cube(edge)

  cube1.display_volume()
  cube1.display_one_surface()
  cube1.display_total_surface()
End Sub
```

7. Solution

```
Class Cube
  Private _edge As Double

  'Define the constructor
  Public Sub New(edge As Double)
    Me.Edge = edge
  End Sub

  'Define public property Edge
  Public Property Edge
    'Define the getter
    Get
      Return Me._edge
    End Get

    'Define the setter
    Set
      If Value > 0 Then
        Me._edge = Value
      Else
        Throw New Exception("Cannot be negative or zero")
      End If
    End Set
  End Property

  Public Sub display_volume()
    Console.WriteLine("Volume: " & Me.Edge ^ 3)
  End Sub

  Public Sub display_one_surface()
    Console.WriteLine("One surface: " & Me.Edge ^ 2)
  End Sub

  Public Sub display_total_surface()
    Console.WriteLine("Total surface: " & 6 * Me.Edge ^ 2)
  End Sub
End Class

Sub Main(args As String())
  Dim edge As Double

  Console.Write("Enter edge length of a cube: ")
```

```
  edge = Console.ReadLine()

  Dim cube1 As New Cube(edge)

  cube1.display_volume()
  cube1.display_one_surface()
  cube1.display_total_surface()
End Sub
```

8. Solution

```
Class Circle
  Private _radius As Double = -1

  'Define public property Radius
  Public Property Radius
    'Define the getter
    Get
      If Me._radius > 0 Then
        Return Me._radius
      Else
        Throw New Exception("Radius is not set")
      End If
    End Get

    'Define the setter
    Set
      If Value > 0 Then
        Me._radius = Value
      Else
        Throw New Exception("Cannot be negative or zero")
      End If
    End Set
  End Property

  Public Function get_diameter() As Double
    Return 2 * Me.Radius
  End Function

  Public Function get_area() As Double
    Return 3.14 * Me.Radius ^ 2
  End Function

  Public Function get_perimeter() As Double
    Return 2 * 3.14 * Me.Radius
  End Function
End Class

Sub display_menu()
  Console.WriteLine("1. Enter radius")
  Console.WriteLine("2. Display radius")
  Console.WriteLine("3. Display diameter")
  Console.WriteLine("4. Display area")
```

```
  Console.WriteLine("5. Display perimeter")
  Console.WriteLine("6. Exit")
End Sub

Sub Main(args As String())
  Dim choice As Integer
  Dim radius As Double

  Dim circle1 As New Circle()

  Do While True
    display_menu()
    Console.Write("Enter a choice: ")
    choice = Console.ReadLine()

    If choice = 6 Then
      Console.WriteLine("Bye")
      Exit Do
    ElseIf choice = 1 Then
      Console.Write("Enter radius: ")
      radius = Console.ReadLine()
      circle1.Radius = radius
    ElseIf choice = 2 Then
      Console.WriteLine("Radius: " & circle1.Radius)
    ElseIf choice = 3 Then
      Console.WriteLine("Diameter: " & circle1.get_diameter())
    ElseIf choice = 4 Then
      Console.WriteLine("Area: " & circle1.get_area())
    ElseIf choice = 5 Then
      Console.WriteLine("Perimeter: " & circle1.get_perimeter())
    End If
  Loop
End Sub
```

9. Solution

```
Class Info
  Private _user_text As String

  'Define public property User_text
  Public Property User_text
    'Define the getter
    Get
      Return Me._user_text
    End Get

    'Define the setter
    Set
      If Value <> "" Then
        Me._user_text = Value
      Else
        Throw New Exception("Cannot be set to empty")
      End If
```

```
    End Set
  End Property

  Public Function get_spaces_count() As Integer
    Dim i, count As Integer
    Dim character As String

    count = 0
    For i = 0 To Me.User_text.Length - 1
      character = Me.User_text(i)
      If character = " " Then
        count += 1
      End If
    Next
    Return count
  End Function

  Public Function get_words_count() As Integer
    Return Me.get_spaces_count() + 1
  End Function

  Public Function get_vowels_count() As Integer
    Dim i, count As Integer
    Dim character As String

    count = 0
    For i = 0 To Me.User_text.Length - 1
      character = Me.User_text.toLower()(i)
      If "aeiou".IndexOf(character) > -1 Then
        count += 1
      End If
    Next
    Return count
  End Function

  Public Function get_letters_count() As Integer
    Return Me.User_text.Length - Me.get_spaces_count()
  End Function
End Class

Sub Main(args As String())
  Dim inf As New Info()

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

  Console.WriteLine("Text: " & inf.User_text)
  Console.WriteLine("Spaces: " & inf.get_spaces_count())
  Console.WriteLine("Words: " & inf.get_words_count())
  Console.WriteLine("Vowels: " & inf.get_vowels_count())
  Console.WriteLine("Total number of letters: " & inf.get_letters_count())
End Sub
```

10. Solution

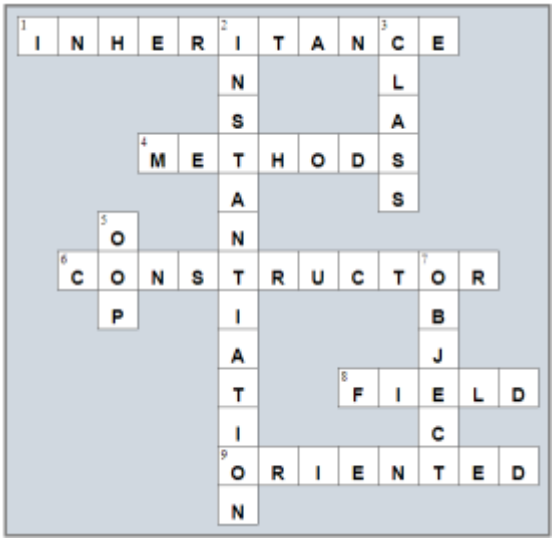
```
Class EncryptDecrypt
  Const alphabet = " abcdefghijklmnopqrstuvwxyz"  'space is a valid character!
  Private _encr_decr_key As Integer = -1

  'Define the propert Encr_decr_key
  Public Property Encr_decr_key
    'Define the getter
    Get
      If Me._encr_decr_key <> -1 Then
        Return Me._encr_decr_key
      Else
        Throw New Exception("Key is not set")
      End If
    End Get

    'Define the setter
    Set
      If Value >= 1 And Value <= 26 Then
        Me._encr_decr_key = Value
      Else
        Throw New Exception("Must be between 1 and 26")
      End If
    End Set
  End Property

  Public Function encrypt(message As String) As String
    Dim character, return_value As String
    Dim new_letter As Char
    Dim i, index, new_index As Integer

    For i = 0 To message.Length - 1
      character = message(i)
      index = alphabet.IndexOf(character)
      new_index = index + Me.Encr_decr_key
      If new_index >= 27 Then
        new_index -= 27
      End If
      new_letter = alphabet(new_index)
      return_value &= new_letter
    Next
    Return return_value
  End Function

  Public Function decrypt(enc_message As String) As String
    Dim character, return_value As String
    Dim new_letter As Char
    Dim i, index, new_index As Integer

    For i = 0 To enc_message.Length - 1
      character = enc_message(i)
```

```
      index = alphabet.IndexOf(character)
      new_index = index - Me.Encr_decr_key
      If new_index < 0 Then
        new_index += 27
      End If
      new_letter = alphabet(new_index)
      return_value &= new_letter
    Next
    Return return_value
  End Function
End Class

Sub display_menu()
  Console.WriteLine("1. Enter encryption/decryption key")
  Console.WriteLine("2. Encrypt a message")
  Console.WriteLine("3. Decrypt a message")
  Console.WriteLine("4. Exit")
End Sub

Sub Main(args As String())
  Dim text As String
  Dim choice As Integer

  Dim ed As New EncryptDecrypt()

  display_menu()
  Console.Write("Enter a choice: ")
  choice = Console.ReadLine()
  Do While choice <> 4
    If choice = 1 Then
      Console.Write("Enter encryption/decryption key: ")
      ed.Encr_decr_key = Console.ReadLine()
    ElseIf choice = 2 Then
      Console.Write("Enter message to encrypt: ")
      text = Console.ReadLine()
      Console.WriteLine("Encrypted message: " & ed.encrypt(text))
    ElseIf choice = 3 Then
      Console.Write("Enter message to decrypt: ")
      text = Console.ReadLine()
      Console.WriteLine("Decrypted message: " & ed.decrypt(text))
    End If

    display_menu()
    Console.Write("Enter a choice: ")
    choice = Console.ReadLine()
  Loop
End Sub
```

Review in “Object Oriented Programming”

Review Crossword Puzzle

1.



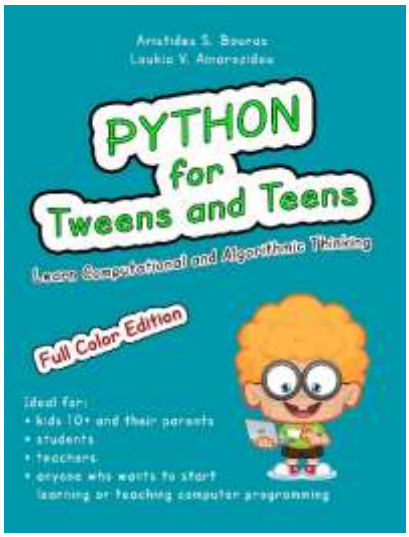
Some Final Words from the Author

I hope you really enjoyed reading this book. I made every possible effort to make it comprehensible even by people that probably have no previous experience in programming.

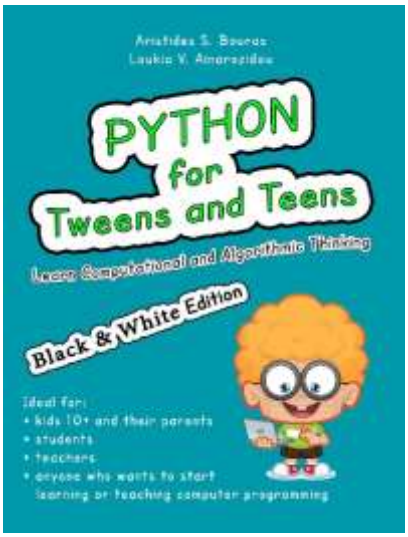
So if you liked this book, please visit the web store where you bought it and show me your gratitude by writing a good review and giving me as many stars as possible. By doing this, you will encourage me to continue writing and of course you'll help other readers to reach me.

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

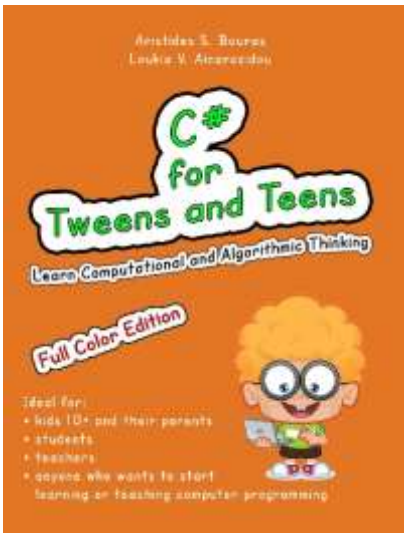
Some of my Books



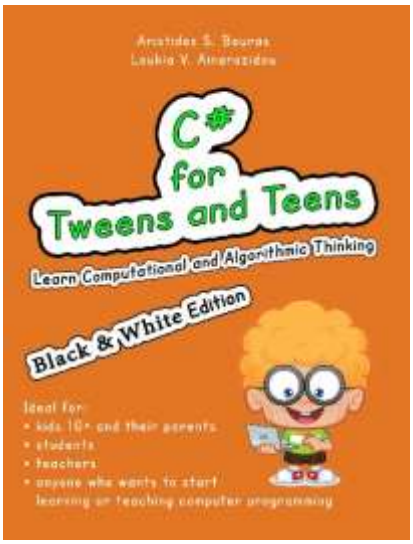
ISBN-10: 1543127940
ISBN-13: 978-1543127942



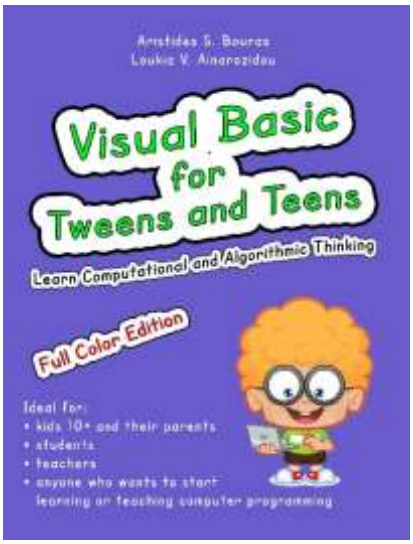
ISBN-10: 1546611215
ISBN-13: 978-1546611219



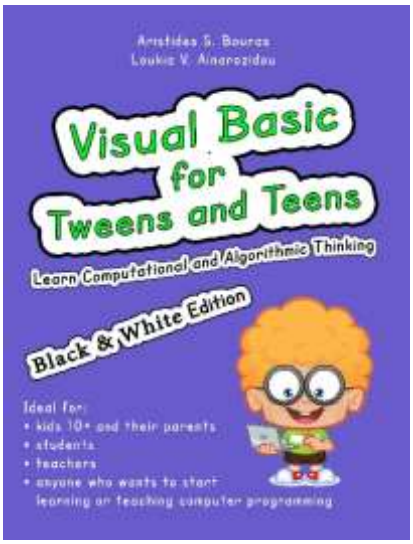
ISBN-10: 1973727684
ISBN-13: 978-1973727682



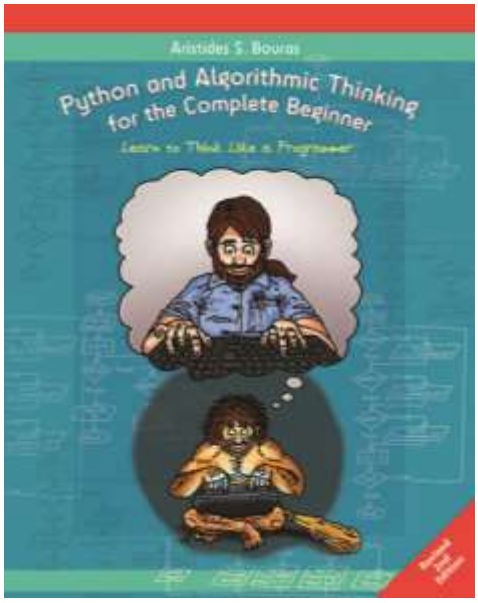
ISBN-10: 1973727765
ISBN-13: 978-1973727767



ISBN-10: 1982083670
ISBN-13: 978-1982083670

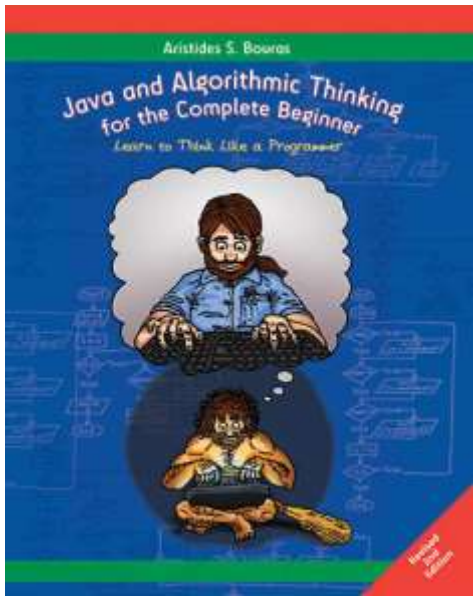


ISBN-10: 1982083697
ISBN-13: 978-1982083694



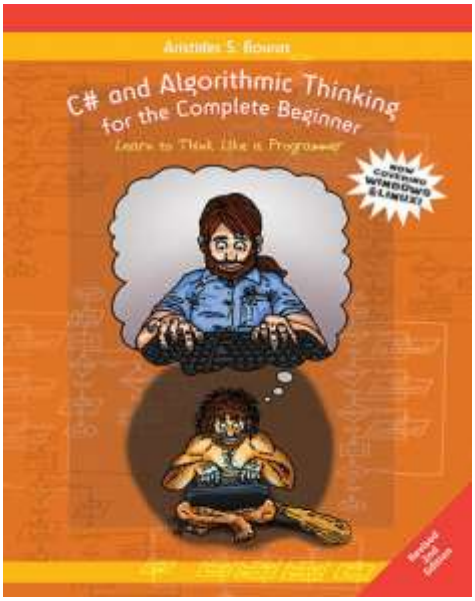
ISBN-10: 1099184878

ISBN-13: 978-1099184871



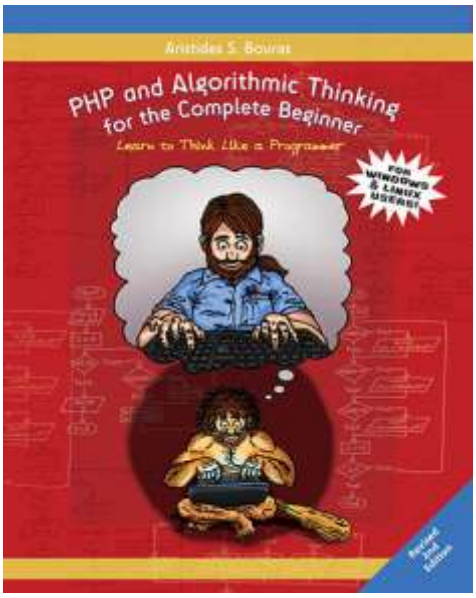
ISBN-10: 1698811233

ISBN-13: 978-1698811239

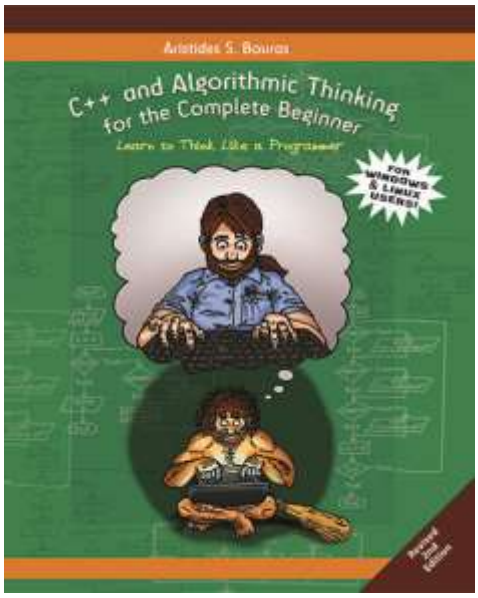


ISBN-10: 1671594363

ISBN-13: 978-1671594364



ISBN-13: 979-8657362534



ISBN-13: 979-8719394534