Java
AND ALGORITHMIC THINKING FOR THE COMPLETE BEGINNER
Second Edition

The Answers

Aristides S. Bouras

Copyright ©

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

RCode: 221008

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

Warning and Disclaimer

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

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

Table of Contents

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

Chapter 12 28
12.2 Review Exercises 28
Chapter 13 31
13.2 Review Exercises 31
Chapter 14 34
14.4 Review Questions: True/False 34
14.5 Review Questions: Multiple Choice 34
14.6 Review Exercises 34
Review in "Sequence Control Structures" 37
Review Crossword Puzzle 37
Chapter 15 38
15.9 Review Questions: True/False 38
15.10 Review Questions: Multiple Choice 38
15.11 Review Exercises 38
Chapter 16 40
16.2 Review Questions: True/False 40
16.3 Review Questions: Multiple Choice 40
16.4 Review Exercises 40
Chapter 17 47
17.2 Review Questions: True/False 47
17.3 Review Questions: Multiple Choice 47
17.4 Review Exercises 47
Chapter 18 55
18.2 Review Questions: True/False 55
18.3 Review Exercises 55
Chapter 19 64
19.2 Review Questions: True/False 64
19.3 Review Exercises 64
Chapter 20 72
20.3 Review Questions: True/False 72
20.4 Review Exercises 72
Chapter 21 78
21.4 Review Exercises 78
Chapter 22 85
22.9 Review Questions: True/False 85

22.10 Review Questions: Multiple Choice 85
22.11 Review Exercises 85
Chapter 23 89
23.6 Review Exercises 89
Review in "Decision Control Structures" 109
Review Crossword Puzzle 109
Chapter 24 110
24.3 Review Questions: True/False 110
Chapter 25 111
25.4 Review Questions: True/False 111
25.5 Review Questions: Multiple Choice 111
25.6 Review Exercises 111
Chapter 26 123
26.3 Review Questions: True/False 123
26.4 Review Questions: Multiple Choice 123
26.5 Review Exercises 123
Chapter 27 137
27.3 Review Questions: True/False 137
27.4 Review Questions: Multiple Choice 137
27.5 Review Exercises 137
Chapter 28 146
28.8 Review Questions: True/False 146
28.9 Review Questions: Multiple Choice 146
28.10 Review Exercises 146
Chapter 29 151
29.4 Review Exercises 151
Chapter 30 160
30.7 Review Questions: True/False 160
30.8 Review Exercises 160
Review in "Loop Control Structures" 184
Review Crossword Puzzle 184
Chapter 31 185
31.13 Review Questions: True/False 185
31.14 Review Questions: Multiple Choice 185
31.15 Review Exercises 185

Chapter 32 199
32.7 Review Questions: True/False 199
32.8 Review Questions: Multiple Choice 199
32.9 Review Exercises 200
Chapter 33 213
33.7 Review Questions: True/False 213
33.8 Review Questions: Multiple Choice 213
33.9 Review Exercises 213
Chapter 34 223
34.7 Review Questions: True/False 223
34.8 Review Exercises 224
Review in "Data Structures in Java" 261
Review Crossword Puzzle 261
Chapter 35 262
35.4 Review Questions: True/False 262
Chapter 36 263
36.8 Review Questions: True/False 263
36.9 Review Exercises 263
Chapter 37 276
37.9 Review Questions: True/False 276
37.10 Review Exercises 276
Chapter 38 287
38.3 Review Exercises 287
Review in "Subprograms" 304
Review Crossword Puzzle 304
Chapter 39 305
39.9 Review Questions: True/False 305
39.10 Review Exercises 305
Review in "Object Oriented Programming" 318
Review Crossword Puzzle 318
Some Final Words from the Author 319

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

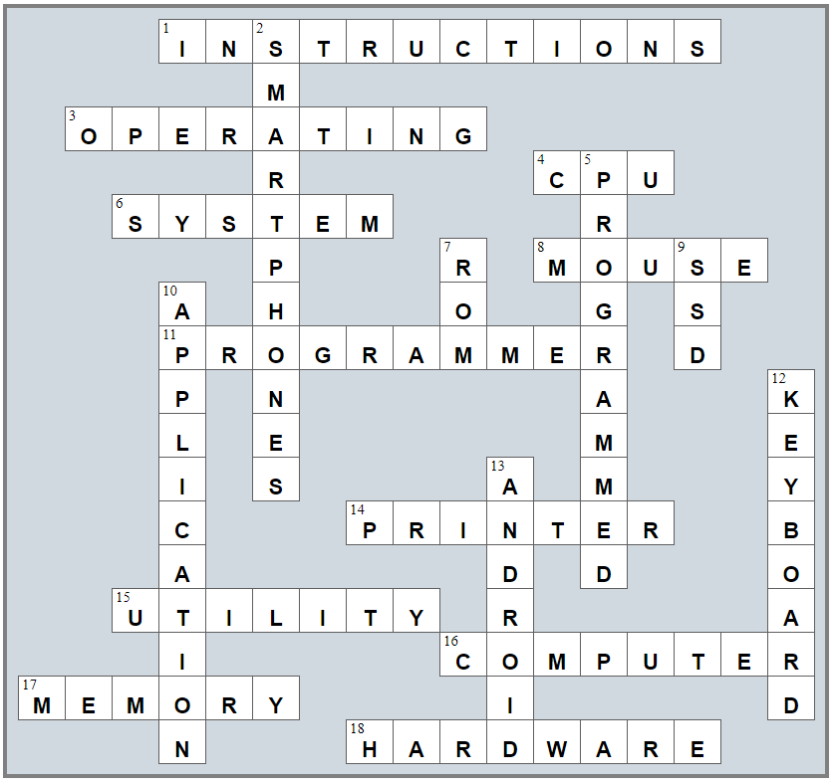
1.8 Review Questions: Multiple Choice

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

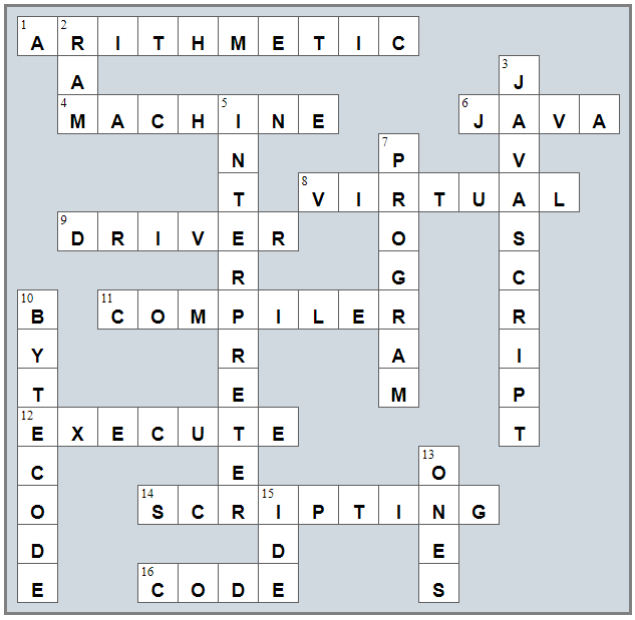
Review in “Introductory Knowledge”

Review Crossword Puzzles

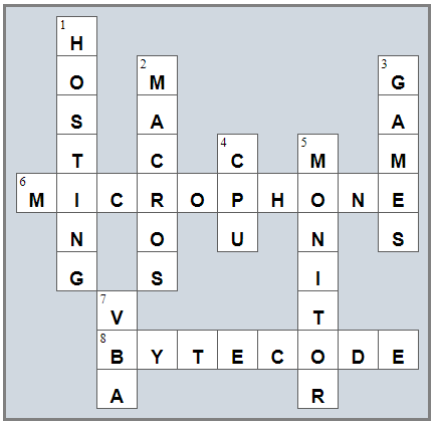
1.



2.



3.


1 H O S T
2 M A C R O S
3 G A M E S
4 C P U
5 M O N I T O R
6 M I N G
C O H O N
7 V
8 B A
Y T E C D E

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

Chapter 6

6.4 Review Questions: True/False

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

6.5 Review Questions: Multiple Choice

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

Chapter 7

7.7 Review Questions: True/False

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

7.8 Review Questions: Multiple Choice

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

7.9 Review Exercises

1. ii, iv, v, ix, x
2. i. String, ii. Boolean, iii. String, iv. String, v. Float, vi. Integer
3. i. d, ii. f, iii. c, iv. e
4. i. 26, ii. 28
5. i. 5, ii. 6
6. i. 1, ii. 0, iii. 1, iv. 1, v. 0, vi. 1
7. i. 2 * 3, ii. 4.0
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 = Integer.parseInt(cin.nextLine())`	**3**	?	?	?
2	`a = (a + 1) * (a + 1) + 6 / 3 * 2 + 20`	**40**	?	?	?
3	`b = a % 13`	40	**1**	?	?
4	`c = b % 7`	40	1	**1**	?
5	`d = a * b * c`	40	1	1	**40**
6	`System.out.println(a + ", " + b + ", " + c + ", " + d)`	It displays: 40, 1, 1, 40			

For the input value of 4

Step	Statement	a	b	c	d
1	`a = Integer.parseInt(cin.nextLine())`	**4**	?	?	?
2	`a = (a + 1) * (a + 1) + 6 / 3 * 2 + 20`	**49**	?	?	?
3	`b = a % 13`	49	**10**	?	?
4	`c = b % 7`	49	10	**3**	?
5	`d = a * b * c`	49	10	3	**1470**
6	`System.out.println(a + ", " + b + ", " + c + ", " + d)`	It displays: 49, 10, 3, 1470			

For the input value of 1

Step	Statement	a	b	c	d
1	`a = Integer.parseInt(cin.nextLine())`	**1**	?	?	?
2	`a = (a + 1) * (a + 1) + 6 / 3 * 2 + 20`	**28**	?	?	?
3	`b = a % 13`	28	**2**	?	?
4	`c = b % 7`	28	2	**2**	?
5	`d = a * b * c`	28	2	2	**112**
6	`System.out.println(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 = Double.parseDouble(cin.nextLine())`	**8.0**	?	?	?	?
2	`b = Double.parseDouble(cin.nextLine())`	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--`	8.0	4.0	78.0	27.0	**38.0**
8	`d -= c + d % c`	8.0	4.0	78.0	**–78.0**	38.0
9	`System.out.println(c + ", " + d + ", " + e)`	It displays: 78.0, –78.0, 38.0				

For the input values of 4, 4

Step	Statement	a	b	c	d	e
1	`a = Double.parseDouble(cin.nextLine())`	**4.0**	?	?	?	?
2	`b = Double.parseDouble(cin.nextLine())`	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--`	4.0	4.0	38.0	11.0	**18.0**
8	`d -= c + d % c`	4.0	4.0	38.0	**–38.0**	18.0
9	`System.out.println(c + ", " + d + ", " + e)`	It displays: 38.0, –38.0, 18.0				

Chapter 9

9.5 Review Exercises

1. Solution

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

2. Solution

For the input values of 5, 5

Step	Statement	a	b	c	d	e
1	`a = Double.parseDouble(cin.nextLine())`	**5.0**	?	?	?	?
2	`b = Double.parseDouble(cin.nextLine())`	5.0	**5.0**	?	?	?
3	`c = a + b`	5.0	5.0	**10.0**	?	?
4	`d = 5 + a / b * c + 2`	5.0	5.0	10.0	**17.0**	?
5	`e = c - d`	5.0	5.0	10.0	17.0	**−7.0**
6	`c += d + c`	5.0	5.0	**37.0**	17.0	−7.0
7	`e--`	5.0	5.0	37.0	17.0	**−8.0**
8	`d += c + a / b`	5.0	5.0	37.0	**55.0**	−8.0
9	`System.out.println(c + ", " + d + ", " + e)`	It displays: 37.0, 55.0, −8.0				

For the input values of 4, 8

Step	Statement	a	b	c	d	e
1	`a = Double.parseDouble(cin.nextLine())`	**4.0**	?	?	?	?
2	`b = Double.parseDouble(cin.nextLine())`	4.0	**8.0**	?	?	?
3	`c = a + b`	4.0	8.0	**12.0**	?	?
4	`d = 5 + a / b * c + 2`	4.0	8.0	12.0	**13.0**	?
5	`e = c - d`	4.0	8.0	12.0	13.0	**−1.0**
6	`c += d + c`	4.0	8.0	**37.0**	13.0	−1.0
7	`e--`	4.0	8.0	37.0	13.0	**−2.0**
8	`d += c + a / b`	4.0	8.0	37.0	**50.5**	−2.0
9	`System.out.println(c + ", " + d + ", " + e)`	It displays: 37.0, 50.5, −2.0				

3. Solution

For the input value of 0.50

Step	Statement	a	b	c
1	`b = Double.parseDouble(cin.nextLine())`	?	**0.50**	?
2	`c = 5`	?	0.50	**5**
3	`c = c * b`	?	0.50	**2.5**
4	`a = 10 * c / 2`	**12.5**	0.50	2.5

5	`System.out.println(a)`	It displays: 12.5

For the input value of 3

Step	Statement	a	b	c
1	`b = Double.parseDouble(cin.nextLine())`	?	**3.0**	?
2	`c = 5`	?	3.0	**5**
3	`c = c * b`	?	3.0	**15.0**
4	`a = 10 * c / 2`	**75.0**	3.0	15.0
5	`System.out.println(a)`	It displays: 75.0		

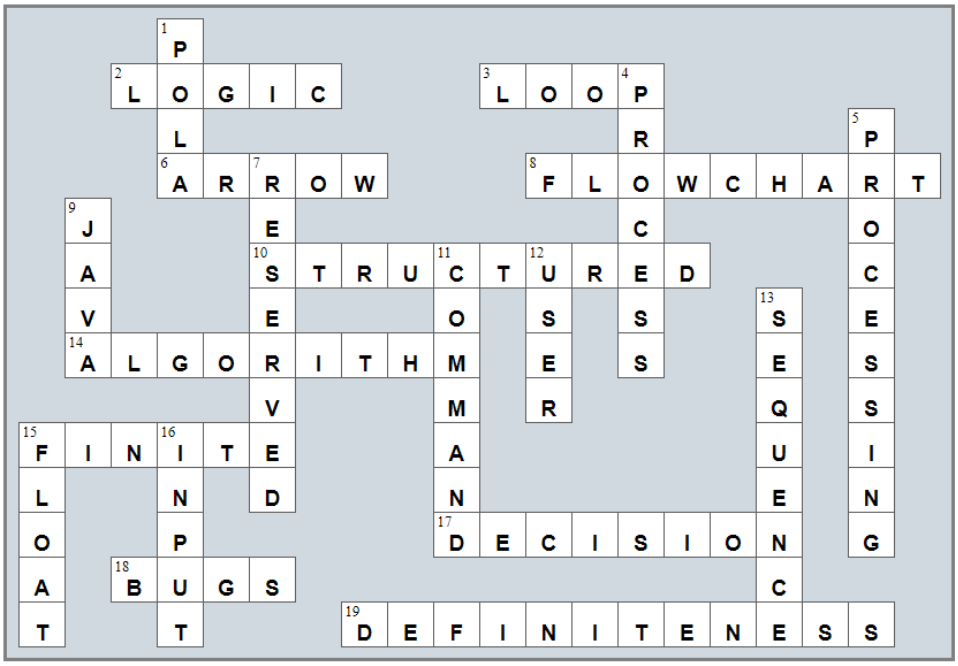
For the input value of 15

Step	Statement	a	b	c
1	`b = Double.parseDouble(cin.nextLine())`	?	**15.0**	?
2	`c = 5`	?	15.0	**5**
3	`c = c * b`	?	15.0	**75.0**
4	`a = 10 * c / 2`	**375.0**	15.0	75.0
5	`System.out.println(a)`	It displays: 375.0		

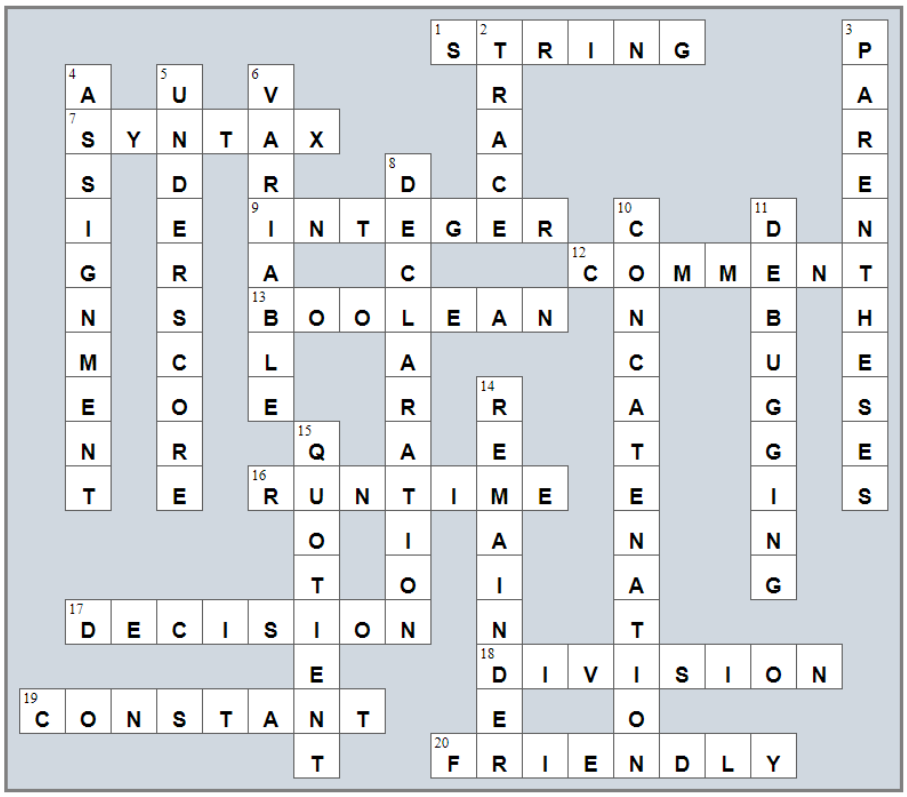
Review in “Getting Started with Java”

Review Crossword Puzzles

1.



2.



Chapter 10

10.2 Review Exercises

1. Solution

```
public static void main(String[] args) {
  double b, h, area;

  System.out.print("Enter base: ");
  b = Double.parseDouble(cin.nextLine());
  System.out.print("Enter height: ");
  h = Double.parseDouble(cin.nextLine());

  area = 0.5 * b * h;

  System.out.println(area);
}
```

2. Solution

```
public static void main(String[] args) {
  double angle1, angle2, angle3;

  System.out.print("Enter 1st angle: ");
  angle1 = Double.parseDouble(cin.nextLine());
  System.out.print("Enter 2nd angle: ");
  angle2 = Double.parseDouble(cin.nextLine());

  angle3 = 180 - angle1 - angle2;

  System.out.println(angle3);
}
```

3. Solution

```
public static void main(String[] args) {
  int g1, g2, g3, g4;
  double average;

  System.out.print("Enter 1st grade: ");
  g1 = Double.parseDouble(cin.nextLine());
  System.out.print("Enter 2nd grade: ");
  g2 = Double.parseDouble(cin.nextLine());
  System.out.print("Enter 3rd grade: ");
  g3 = Double.parseDouble(cin.nextLine());
  System.out.print("Enter 4th grade: ");
  g4 = Double.parseDouble(cin.nextLine());

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

```
  System.out.println(average);
}
```

4. Solution

```
static final double PI = 3.14159;

public static void main(String[] args) {
  double r, perimeter;

  System.out.print("Enter radius: ");
  r = Double.parseDouble(cin.nextLine());

  perimeter = 2 * PI * r;

  System.out.println(perimeter);
}
```

5. Solution

```
static final double PI = 3.14159;

public static void main(String[] args) {
  double d, radius, volume;

  System.out.print("Enter diameter (in meters): ");
  d = Double.parseDouble(cin.nextLine());

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

  System.out.println(volume);
}
```

6. Solution

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

7. Solution

```
static final double PI = 3.14159;

public static void main(String[] args) {
  double d, radius, perimeter, area, volume;

  System.out.print("Enter diameter: ");
  d = Double.parseDouble(cin.nextLine());

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

```
  System.out.println(radius + " " + perimeter + " " + area + " " + volume);
}
```

8. Solution

```
public static void main(String[] args) {
  double charge, tip, tax, total;

  System.out.print("Enter charge for a meal: ");
  charge = Double.parseDouble(cin.nextLine());

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

  total = charge + tip + tax;

  System.out.println(total);
}
```

9. Solution

```
public static void main(String[] args) {
  double a, t, s;

  System.out.print("Enter acceleration in m/sec2: ");
  a = Double.parseDouble(cin.nextLine());
  System.out.print("Enter time traveled in sec: ");
  t = Double.parseDouble(cin.nextLine());

  s = 0.5 * a * t * t;

  System.out.println(s);
}
```

10. Solution

```
public static void main(String[] args) {
  double f, c;

  System.out.print("Enter temperature in Fahrenheit: ");
  f = Double.parseDouble(cin.nextLine());

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

  System.out.println(c);
}
```

11. Solution

```
public static void main(String[] args) {
  int w, h;
```

```
  double bmi;

  System.out.print("Enter weight in pounds: ");
  w = Integer.parseInt(cin.nextLine());
  System.out.print("Enter height in inches: ");
  h = Integer.parseInt(cin.nextLine());

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

  System.out.println(bmi);
}
```

12. Solution

```
public static void main(String[] args) {
  double s_total, g_rate, tip, total;

  System.out.print("Enter subtotal: ");
  s_total = Double.parseDouble(cin.nextLine());
  System.out.print("Enter gratuity rate (0 - 100): ");
  g_rate = Double.parseDouble(cin.nextLine());

  tip = s_total * g_rate / 100;

  total = s_total + tip;

  System.out.println("Tip is $" + tip + " and total is $" + total);
}
```

13. Solution

```
static final double VAT = 0.20;

public static void main(String[] args) {
  double btax_price1, btax_price2, btax_price3, atax_price1, atax_price2, atax_price3, avg;

  System.out.print("Enter before-tax price 1: ");
  btax_price1 = Double.parseDouble(cin.nextLine());
  System.out.print("Enter before-tax price 2: ");
  btax_price2 = Double.parseDouble(cin.nextLine());
  System.out.print("Enter before-tax price 3: ");
  btax_price3 = Double.parseDouble(cin.nextLine());

  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;

  System.out.println(avg);
}
```

14. Solution

```
static final int VAT = 0.20;

public static void main(String[] args) {
  double atax_price, btax_price;

  System.out.print("Enter after-tax price: ");
  atax_price = Integer.parseInt(cin.nextLine());

  btax_price = atax_price / (1 + VAT);

  System.out.println(btax_price);
}
```

15. Solution

```
public static void main(String[] args) {
  double i_price, discount, f_price, saved;

  System.out.print("Enter price: ");
  i_price = Double.parseDouble(cin.nextLine());
  System.out.print("Enter discount: ");
  discount = Double.parseDouble(cin.nextLine());

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

  System.out.println(f_price + " " + saved);
}
```

16. Solution

```
static final int VAT = 0.20;

public static void main(String[] args) {
  int i_kWh, f_kWh, kWh_consumed;
  double cost;

  System.out.print("Enter kWh at the beginning of the month: ");
  i_kWh = Integer.parseInt(cin.nextLine());
  System.out.print("Enter kWh at the end of the month: ");
  f_kWh = Integer.parseInt(cin.nextLine());

  kWh_consumed = f_kWh - i_kWh;

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

  System.out.println(kWh_consumed + " " + cost);
}
```

17. Solution

```
public static void main(String[] args) {
  int day, month, days_passed, days_left;

  System.out.print("Enter current month: ");
  month = Integer.parseInt(cin.nextLine());
  System.out.print("Enter current day: ");
  day = Integer.parseInt(cin.nextLine());

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

  System.out.println(days_left);
}
```

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 = Double.parseDouble(cin.nextLine())`	**9.0**	?	?
2	`a += 6 / Math.sqrt(a) * 2 + 20.4`	**33.4**	?	**?**
3	`b = Math.round(a) % 4`	33.4	**1.0**	**?**
4	`c = b % 3`	33.4	1.0	**1.0**
5	`System.out.println(a + ", " + b + ", " + c)`	It displays: 33.4, 1.0, 1.0		

For the input value of 4

Step	Statement	a	b	c
1	`a = Double.parseDouble(cin.nextLine())`	**4.0**	?	?
2	`a += 6 / Math.sqrt(a) * 2 + 20.4`	**30.4**	?	**?**
3	`b = Math.round(a) % 4`	30.4	**2.0**	**?**
4	`c = b % 3`	30.4	2.0	**2.0**
5	`System.out.println(a + ", " + b + ", " + c)`	It displays: 30.4, 2.0, 2.0		

2. Solution

For the input value of −2

Step	Statement	a	b	c
1	`a = Integer.parseInt(cin.nextLine())`	**−2**	?	?
2	`b = Math.abs(a) % 4 + Math.pow(a, 4)`	−2	**18**	**?**
3	`c = b % 5`	−2	18	**3**
4	`System.out.println(b + ", " + c)`	It displays: 18, 3		

For the input value of –3

Step	Statement	a	b	c
1	`a = Integer.parseInt(cin.nextLine())`	**–3**	?	?
2	`b = Math.abs(a) % 4 + Math.pow(a, 4)`	–3	**84**	**?**
3	`c = b % 5`	–3	84	**4**
4	`System.out.println(b + ", " + c)`	It displays: 84, 4		

3. Solution

```
public static void main(String[] args) {
  double degrees, radians;

  System.out.print("Enter angle in radians: ");
  radians = Double.parseDouble(cin.nextLine());

  degrees = radians * 180 / Math.PI;

  System.out.println(degrees);
}
```

4. Solution

```
public static void main(String[] args) {
  double a, b, hypotenuse;

  System.out.print("Enter right angle side A of a right-angled triangle: ");
  a = Double.parseDouble(cin.nextLine());
  System.out.print("Enter right angle side B of a right-angled triangle: ");
  b = Double.parseDouble(cin.nextLine());

  hypotenuse = Math.sqrt(Math.pow(a, 2) + Math.pow(b, 2));

  System.out.println(hypotenuse);
}
```

5. Solution

```
public static void main(String[] args) {
  double adjacent, opposite, th;

  System.out.print("Enter angle θ (in degrees) of a right-angled triangle: ");
  th = Double.parseDouble(cin.nextLine());
  System.out.print("Enter length of adjacent side: ");
  adjacent = Double.parseDouble(cin.nextLine());

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

  System.out.println(opposite);
}
```

Chapter 12

12.2 Review Exercises

1. Solution

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

2. Solution

i. `y = Math.pow(x + 3, 5 * w) / (7 * (x - 4))`
ii. `y = Math.pow(3 * Math.pow(x, 2) - Math.pow(x, 3) / 4, 1 / 5.0)`
iii. `y = Math.sqrt(Math.pow(x, 4) - 2 * Math.pow(x, 3) - 7 * x * x +x) / Math.pow(4 * (7 * Math.pow(x, 4) - 3 / 4.0 * Math.pow(x, 3)) * (7 * x * x + x), 1/3.0)`
iv. `y = x / (x - 3 * (x - 1)) + x * Math.pow(x - 1, 1 / 5.0) / ((Math.pow(x, 3) - 2) * Math.pow(x - 1, 3))`
v. `y = Math.pow(Math.sin(Math.PI / 3) - Math.cos(Math.PI / 2 * w) , 2)`
vi. `y = Math.pow(Math.sin(Math.PI / 2 * x) + Math.cos(3 * Math.PI / 2 * w), 3) / Math.pow(Math.tan(2 * Math.PI / 3 * w) - Math.sin(Math.PI / 2 * x), 0.5) + 6`

3. Solution

```
public static void main(String[] args) {
  double x, y;

  System.out.print("Enter value for x: ");
  x = Double.parseDouble(cin.nextLine());

  y = Math.sqrt(x) * (Math.pow(x, 3) + Math.pow(x, 2));

  System.out.println(y);
}
```

4. Solution

```
public static void main(String[] args) {
  double x, y;

  System.out.print("Enter value for x: ");
  x = Double.parseDouble(cin.nextLine());

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

  System.out.println(y);
}
```

5. Solution

```
public static void main(String[] args) {
  double w, x, y;
```

```
  System.out.print("Enter value for x: ");
  x = Double.parseDouble(cin.nextLine());
  System.out.print("Enter value for w: ");
  w = Double.parseDouble(cin.nextLine());

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

  System.out.println(y);
}
```

6. Solution

```
public static void main(String[] args) {
  double w, x, y;

  System.out.print("Enter value for x: ");
  x = Double.parseDouble(cin.nextLine());
  System.out.print("Enter value for w: ");
  w = Double.parseDouble(cin.nextLine());

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

  System.out.println(y);
}
```

7. Solution

```
public static void main(String[] args) {
  double w, x, y;

  System.out.print("Enter value for x: ");
  x = Double.parseDouble(cin.nextLine());
  System.out.print("Enter value for w: ");
  w = Double.parseDouble(cin.nextLine());

  y = Math.pow(x, x) / Math.pow(Math.sin(2 * w / 3 + 5) - x, 2) + Math.pow(Math.sin(3 * x) + w, x + 1)
/ Math.pow(Math.sqrt(7 * w), 3 / 2);

  System.out.println(y);
}
```

8. Solution

```
public static void main(String[] args) {
  double a, b, c, area, semi;

  System.out.print("Enter length A: ");
  a = Double.parseDouble(cin.nextLine());
  System.out.print("Enter length B: ");
  b = Double.parseDouble(cin.nextLine());
```

```
  System.out.print("Enter length C: ");
  c = Double.parseDouble(cin.nextLine());

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

  System.out.println(area);
}
```

Chapter 13

13.2 Review Exercises

1. Solution

```
public static void main(String[] args) {
  int last_digit, n, result;

  System.out.print("Enter an integer: ");
  n = Integer.parseInt(cin.nextLine());

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

  System.out.println(result);
}
```

2. Solution

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

  System.out.print("Enter a five-digit integer: ");
  number = Integer.parseInt(cin.nextLine());

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

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

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

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

  reversed_number = digit5 * 10000 + digit4 * 1000 + digit3 * 100 + digit2 * 10 + digit1;
  System.out.println(reversed_number);
}
```

3. Solution

```
public static void main(String[] args) {
  int n, result;

  System.out.print("Enter an integer: ");
  n = Integer.parseInt(cin.nextLine());

  result = n % 2;
```

```
  System.out.println(result);
}
```

4. Solution

```
public static void main(String[] args) {
  int n, result;

  System.out.print("Enter an integer: ");
  n = Integer.parseInt(cin.nextLine());

  result = 1 - n % 2;

  System.out.println(result);
}
```

5. Solution

```
public static void main(String[] args) {
  int days, hours, minutes, number, r, seconds, weeks;

  System.out.print("Enter an elapsed time in seconds: ");
  number = Integer.parseInt(cin.nextLine());

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

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

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

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

  System.out.print(weeks + " week(s) " + days + " day(s) " + hours + " hour(s) ");
  System.out.println(minutes + " minute(s) and " + seconds + " second(s)");
}
```

6. Solution

```
public static void main(String[] args) {
  int amount, r, usd1, usd10, usd20, usd5;

  System.out.print("Enter amount of money to withdraw: ");
  amount = Integer.parseInt(cin.nextLine());

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

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

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

  System.out.print(usd20 + " note(s) of $20 " + usd10 + " note(s) of $10 ");
  System.out.println(usd5 + " note(s) of $5 and " + usd1 + " note(s) of $1");
}
```

7. Solution

```
public static void main(String[] args) {
  int distance, feet, inches, miles, r, steps, yards;

  System.out.print("Enter number of steps: ");
  steps = Integer.parseInt(cin.nextLine());

  distance = steps * 25;

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

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

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

  System.out.print(miles + " mile(s) " + yards + " yard(s) ");
  System.out.println(feet + " foot/feet and " + inches + " inch(es)");
}
```

Chapter 14

14.4 Review Questions: True/False

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

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

```
public static void main(String[] args) {
  String first_name, last_name, middle_name, title;

  System.out.print("First name: ");
  first_name = cin.nextLine();
  System.out.print("Middle name: ");
  middle_name = cin.nextLine();
  System.out.print("Last name: ");
  last_name = cin.nextLine();
  System.out.print("Title: ");
  title = cin.nextLine();

  System.out.println(title + " " + first_name + " " + middle_name + " " + last_name);
  System.out.println(first_name + " " + middle_name + " " + last_name);
  System.out.println(last_name + ", " + first_name);
  System.out.println(last_name + ", " + first_name + " " + middle_name);
  System.out.println(last_name + ", " + first_name + " " + middle_name + ", " + title);
  System.out.println(first_name + " " + last_name);
}
```

2. Solution

```
public static void main(String[] args) {
  String alphabet, random_word;

  alphabet = "abcdefghijklmnopqrstuvwxyz";

  random_word = ("" + alphabet.charAt((int)(Math.random() * 26))).toUpperCase() +
                      alphabet.charAt((int)(Math.random() * 26)) +
                      alphabet.charAt((int)(Math.random() * 26)) +
                      alphabet.charAt((int)(Math.random() * 26)) +
```

```
                        alphabet.charAt((int)(Math.random() * 26));

  System.out.println(random_word);
}
```

3. Solution

```
public static void main(String[] args) {
  String name, secret_password;

  System.out.print("Enter name: ");
  name = cin.nextLine().toLowerCase().replace(" ", "");

  secret_password = "" + name.charAt((int)(Math.random() * name.length())) +
                         name.charAt((int)(Math.random() * name.length())) +
                         name.charAt((int)(Math.random() * name.length())) +
                         (1000 + (int)(Math.random() * (9999 - 1000 + 1)));

  System.out.println(secret_password);
}
```

4. Solution

First approach

```
public static void main(String[] args) {
  int number, reversed_number;
  String s_number, digit1, digit2, digit3;

  System.out.print("Enter an three-digit integer: ");
  number = Integer.parseInt(cin.nextLine());

  s_number = "" + number;

  digit1 = "" + s_number.charAt(0);
  digit2 = "" + s_number.charAt(1);
  digit3 = "" + s_number.charAt(2);

  reversed_number = 100 * Integer.parseInt(digit3) +
                    10 * Integer.parseInt(digit2) + Integer.parseInt(digit1);

  System.out.println(reversed_number);
}
```

First approach

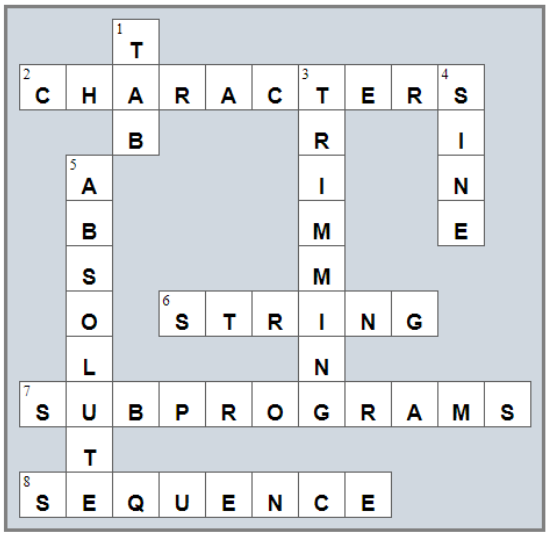
```
public static void main(String[] args) {
  int number, reversed_number;
  String s_number;

  System.out.print("Enter an three-digit integer: ");
  number = Integer.parseInt(cin.nextLine());

  s_number = "" + number;
```

```
        reversed_number = Integer.parseInt("" + s_number.charAt(2) +
                                            s_number.charAt(1) + s_number.charAt(0));
  System.out.println(reversed_number);
}
```

Review in “Sequence Control Structures”

Review Crossword Puzzle

1.



Chapter 15

15.9 Review Questions: True/False

1. true
2. false
3. false
4. false
5. false
6. false
7. false
8. true
9. true
10. true
11. true
12. true
13. true
14. true
15. 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 || BE2	BE1 && BE2	!(BE2)
false	**false**	false	false	true
false	**true**	true	false	false
true	**false**	true	false	true
true	**true**	true	true	false

4. Solution

a	b	c	a > 3 || c > b && c > 1	a > 3 && c > b || c > 1
4	**–6**	**2**	true	true

−3	**2**	**−4**	false	false
2	**5**	**5**	false	true

5. Solution

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

6. Solution

i. false
ii. true

7. Solution

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

8. Solution

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

9. Solution

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

Chapter 16

16.2 Review Questions: True/False

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

16.3 Review Questions: Multiple Choice

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

16.4 Review Exercises

1. Solution

The corrections/additions are in red

```
public static void main(String[] args) {
  double x, y;

  x = Double.parseDouble(cin.nextLine());

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

  System.out.println(x + " " + y);
}
```

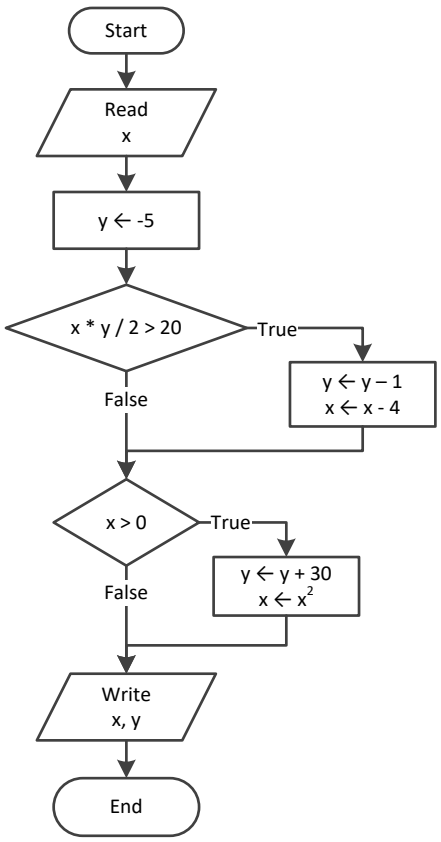
2. Solution

For the input value of 10

Step	Statement	x	y
1	`x = Double.parseDouble(cin.nextLine())`	**10.0**	?
2	`y = -5`	10.0	**−5.0**
3	`if (x * y / 2 > 20)`	false	
4	`if (x > 0)`	true	
5	`y += 30`	10.0	**25.0**
6	`x = Math.pow(x, 2)`	**100.0**	25.0
7	`System.out.println(x + ", " + y)`	It displays: 100.0, 25.0	

For the input value of −10

Step	Statement	x	y
1	`x = Double.parseDouble(cin.nextLine())`	**−10.0**	?
2	`y = -5`	−10.0	**−5.0**
3	`if (x * y / 2 > 20)`	true	
4	`y--`	−10.0	**−6.0**
5	`x -= 4`	**−14.0**	−6.0
6	`if (x > 0)`	false	
7	`System.out.println(x + ", " + y)`	It displays: −14.0, −6.0	



3. *Solution*

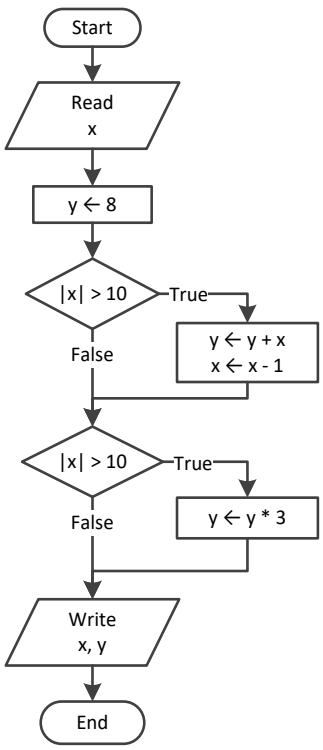
For the input value of −11

Step	Statement	x	y
1	`x = Integer.parseInt(cin.nextLine())`	**−11**	?
2	`y = 8`	−11	**8**
3	`if (Math.abs(x) > 10)`	true	
4	`y += x`	−11	**−3**
5	`x--`	**−12**	−3
6	`if (Math.abs(x) > 10)`	true	
7	`System.out.println(x + ", " + y)`	−12	**−9**

8	`x = Integer.parseInt(cin.nextLine())`	It displays: –12, –9

For the input value of 11

Step	Statement	x	y
1	`x = Integer.parseInt(cin.nextLine())`	**11**	?
2	`y = 8`	11	**8**
3	`if (Math.abs(x) > 10)`	true	
4	`y += x`	11	**19**
5	`x--`	**10**	19
6	`if (Math.abs(x) > 10)`	false	
7	`System.out.println(x + ", " + y)`	It displays: 10, 19	



4. *Solution*

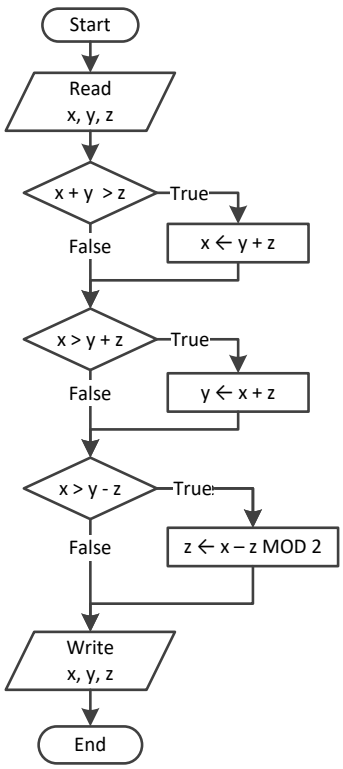
For input values of 1, 2 and 3

Step	Statement	x	y	z
1	`x = Integer.parseInt(cin.nextLine())`	**1**	?	?
2	`y = Integer.parseInt(cin.nextLine())`	1	**2**	?
3	`z = Integer.parseInt(cin.nextLine())`	1	**2**	**3**
4	`if (x + y > z)`	false		
5	`if (x > y + z)`	false		
6	`if (x > y - z)`	true		
7	`z = x - z % 2`	1	2	**0**

8	`System.out.println(x + ", " + y + ", " + z)`	It displays: 1, 2, 0

For input values of 4, 2 and 1

Step	Statement	x	y	z
1	`x = Integer.parseInt(cin.nextLine())`	**4**	?	?
2	`y = Integer.parseInt(cin.nextLine())`	4	**2**	?
3	`z = Integer.parseInt(cin.nextLine())`	4	**2**	**1**
4	`if (x + y > z)`	true		
5	`x = y + z`	**3**	2	1
6	`if (x > y + z)`	false		
7	`if (x > y - z)`	true		
8	`z = x - z % 2`	3	2	**2**
9	`System.out.println(x + ", " + y + ", " + z)`	It displays: 3, 2, 2		



5. Solution

```
public static void main(String[] args) {
  double x;

  System.out.print("Enter a number: ");
  x = Double.parseDouble(cin.nextLine());

  if (x > 0) {
    System.out.println("Positive");
  }
```

```
}
```

6. Solution

```
public static void main(String[] args) {
  double x, y;

  System.out.print("Enter a number: ");
  x = Double.parseDouble(cin.nextLine());
  System.out.print("Enter a second number");
  y = Double.parseDouble(cin.nextLine());

  if (x > 0 && y > 0) {
    System.out.println("Positive");
  }
}
```

7. Solution

```
public static void main(String[] args) {
  double x, y;

  System.out.print("Enter your age: ");
  x = Double.parseDouble(cin.nextLine());

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

8. Solution

```
public static void main(String[] args) {
  String str;

  System.out.print("Enter a string: ");
  str = cin.nextLine();

  if (str.equals(str.toUpperCase()) == true) {
    System.out.println("Uppercase");
  }
}
```

9. Solution

```
public static void main(String[] args) {
  String str;

  System.out.print("Enter a string: ");
  str = cin.nextLine();

  if (str.length() > 20) {
```

```
    System.out.println("Many characters");
  }
}
```

10. Solution

```
public static void main(String[] args) {
  double n1, n2, n3, n4;

  System.out.print("Enter 1st number: ");
  n1 = Double.parseDouble(cin.nextLine());
  System.out.print("Enter 2nd number: ");
  n2 = Double.parseDouble(cin.nextLine());
  System.out.print("Enter 3rd number: ");
  n3 = Double.parseDouble(cin.nextLine());
  System.out.print("Enter 4th number: ");
  n4 = Double.parseDouble(cin.nextLine());

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

11. Solution

```
public static void main(String[] args) {
  double a, b, c;

  System.out.print("Enter 1st number: ");
  a = Double.parseDouble(cin.nextLine());
  System.out.print("Enter 2nd number: ");
  b = Double.parseDouble(cin.nextLine());

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

  System.out.println(a + ", " + b);
}
```

12. Solution

```
public static void main(String[] args) {
  double average, t1, t2, t3;

  System.out.print("Enter 1st temperature: ");
  t1 = Double.parseDouble(cin.nextLine());
  System.out.print("Enter 2nd temperature: ");
  t2 = Double.parseDouble(cin.nextLine());
  System.out.print("Enter 3rd temperature: ");
```

```
  t3 = Double.parseDouble(cin.nextLine());

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

  if (average > 60) {
    System.out.println("Heat Wave");
  }
}
```

Chapter 17

17.2 Review Questions: True/False

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

17.3 Review Questions: Multiple Choice

1. b
2. c
3. c

17.4 Review Exercises

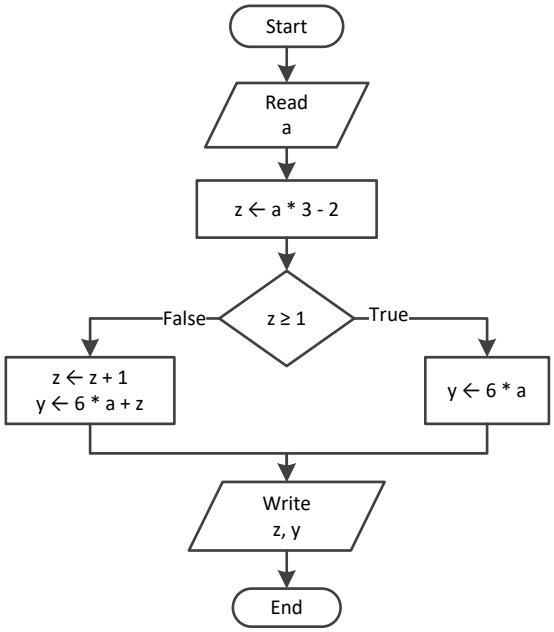
1. Solution

For input value of 3

Step	Statement	a	y	z
1	`a = Double.parseDouble(cin.nextLine())`	**3.0**	?	?
2	`z = a * 3 - 2`	3.0	?	**7.0**
3	`if (z >= 1)`		true	
4	`y = 6 * a`	3.0	**18.0**	7.0
5	`System.out.println(z + ", " + y)`	It displays: 7.0 18.0		

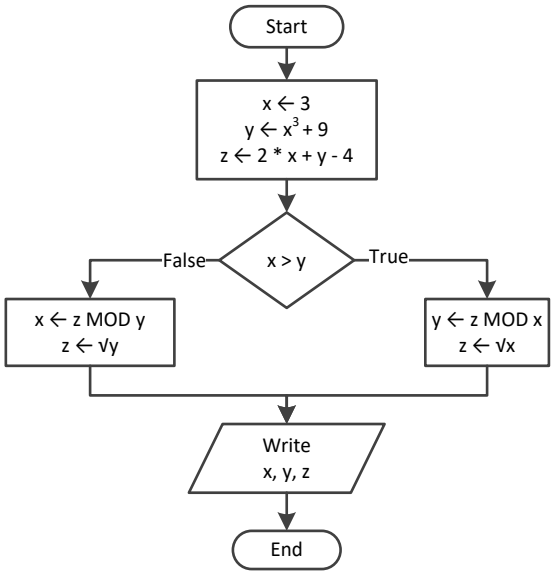
For input value of 0.5

Step	Statement	a	y	z
1	`a = Double.parseDouble(cin.nextLine())`	**0.5**	?	?
2	`z = a * 3 - 2`	0.5	?	**–0.5**
3	`if (z >= 1)`		false	
4	`z++`	0.5	?	**0.5**
5	`y = 6 * a + z`	0.5	**3.5**	0.5
6	`System.out.println(z + ", " + y)`	It displays: 0.5, 3.5		



2. Solution

Step	Statement	x	y	z
1	`x = 3`	**3.0**	?	?
2	`y = Math.pow(x, 3) + 9`	3.0	**36.0**	?
3	`z = 2 * x + y - 4`	3.0	36.0	**38.0**
4	`if (x > y)`		false	
5	`x = z % y`	**2.0**	36.0	38.0
6	`z = Math.sqrt(y)`	2.0	36.0	**6.0**
7	`System.out.println(x + ", " + y + ", " + z)`	It displays: 2.0, 36.0, 6.0		



3. Solution

```
public static void main(String[] args) {
  double w, x, y, z;

  x = Double.parseDouble(cin.nextLine());
  w = x * 3 - 15;
  z = (w + 7) * (x + 4) - 10;
  if (w > x && z > x) {
    x++;
    y = x / 2 + 4;
  }
  else {
    y = x / 4 + 2;
  }
  System.out.println(y);
}
```

For input value of 10

Step	Statement	x	y	w	z
1	`x = Double.parseDouble(cin.nextLine())`	**10.0**	?	?	?
2	`w = x * 3 - 15`	10.0	?	**15.0**	?
3	`z = (w + 7) * (x + 4) - 10`	10.0	?	15.0	**298.0**
4	`if (w > x && z > x)`	true			
5	`x++`	**11.0**	?	15.0	298.0
6	`y = x / 2 + 4`	11.0	**9.5**	15.0	298.0
7	`System.out.println(y)`	It displays: 9.5			

For input value of 2

Step	Statement	x	y	w	z
1	`x = Double.parseDouble(cin.nextLine())`	**2.0**	?	?	?
2	`w = x * 3 - 15`	2.0	?	**–9.0**	?
3	`z = (w + 7) * (x + 4) - 10`	2.0	?	–9.0	**–22.0**
4	`if (w > x && z > x)`	false			
5	`y = x / 4 + 2`	2.0	**2.5**	–9.0	–22.0
6	`System.out.println(y)`	It displays: 2.5			

4. Solution

```
public static void main(String[] args) {
  String name1, name2;
  int goals1, goals2;

  System.out.print("Enter team name 1: ");
  name1 = cin.nextLine();
  System.out.print("Enter team name 2: ");
  name2 = cin.nextLine();
```

```
  System.out.print("Enter goals " + name1 + " scored: ");
  goals1 = Integer.parseInt(cin.nextLine());
  System.out.print("Enter goals " + name2 + " scored: ");
  goals2 = Integer.parseInt(cin.nextLine());

  if (goals1 > goals2) {
    System.out.println("Winner: " + name1);
  }
  else {
    System.out.println("Winner: " + name2);
  }
}
```

5. Solution

```
public static void main(String[] args) {
  int x;

  x = Integer.parseInt(cin.nextLine());
  if (x % 6 == 0) {
    System.out.println(x + " is a multiple of 6");
  }
  else {
    System.out.println(x + " is not a multiple of 6");
  }
}
```

6. Solution

```
public static void main(String[] args) {
  int x;

  x = Integer.parseInt(cin.nextLine());
  if (x % 6 == 0 || x % 7 == 0) {
    System.out.println(x + " is a multiple of 6 or a multiple of 7");
  }
  else {
    System.out.println(x + " is neither a multiple of 6 nor a multiple of 7");
  }
}
```

7. Solution

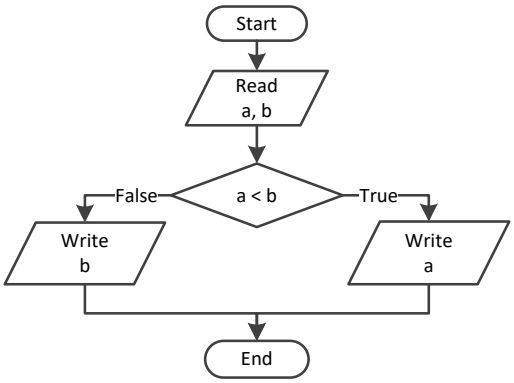
```
public static void main(String[] args) {
  int x, y;

  x = Integer.parseInt(cin.nextLine());

  y = x % 4;
  if (y == 0) {
    System.out.println(x + " is a multiple of 4");
```

```
  }
  else {
    System.out.println(x + " is not a multiple of 4");
  }

  System.out.println("The structure is: " + x + " = " + (int)(x / 4) + " x 4 + " + y);
}
```

8. Solution

```
public static void main(String[] args) {
  int x;

  x = Integer.parseInt(cin.nextLine());

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

9. Solution




```
public static void main(String[] args) {
  double a, b;

  a = Double.parseDouble(cin.nextLine());
  b = Double.parseDouble(cin.nextLine());

  if (a < b) {
    System.out.println(a);
  }
  else {
    System.out.println(b);
  }
}
```

10. Solution

```
public static void main(String[] args) {
  double a, b, c;

  a = Double.parseDouble(cin.nextLine());
  b = Double.parseDouble(cin.nextLine());
  c = Double.parseDouble(cin.nextLine());

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

11. Solution

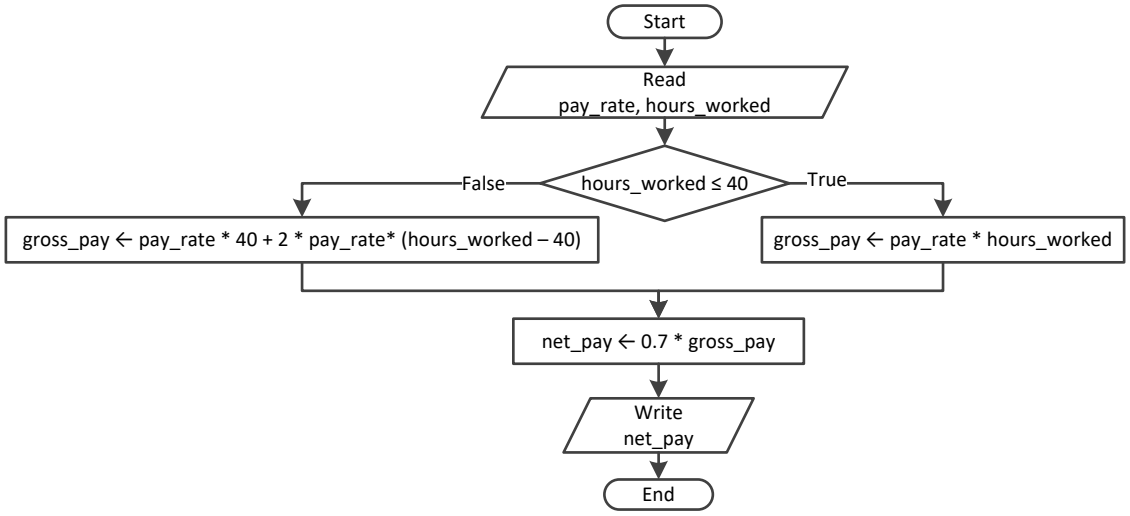
```
public static void main(String[] args) {
  double a, b, c;

  a = Double.parseDouble(cin.nextLine());
  b = Double.parseDouble(cin.nextLine());
  c = Double.parseDouble(cin.nextLine());

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

12. Solution

```
public static void main(String[] args) {
  double a, average, b, c;

  System.out.print("Enter 1st jump in meters: ");
  a = Double.parseDouble(cin.nextLine());
  System.out.print("Enter 2nd jump in meters: ");
  b = Double.parseDouble(cin.nextLine());
  System.out.print("Enter 3rd jump in meters: ");
  c = Double.parseDouble(cin.nextLine());

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

  if (average >= 8) {
```

```
    System.out.println("Qualified");
  }
  else {
    System.out.println("Disqualified");
  }
}
```

13. Solution




```
public static void main(String[] args) {
  double gross_pay, net_pay, pay_rate;
  int hours_worked;

  pay_rate = Double.parseDouble(cin.nextLine());
  hours_worked = Integer.parseInt(cin.nextLine());

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

  net_pay = 0.7 * gross_pay;
  System.out.println(net_pay);
}
```

14. Solution

```
public static void main(String[] args) {
  int miles, miles_left, r;

  System.out.print("Enter miles traveled: ");
  miles = Integer.parseInt(cin.nextLine());

  r = miles % 12000;
```

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

15. Solution

```
public static void main(String[] args) {
  double a1, a2, s1, s2, t;

  System.out.print("Enter the time the two cars traveled: ");
  t = Double.parseDouble(cin.nextLine());
  System.out.print("Enter the accelaration for car A: ");
  a1 = Double.parseDouble(cin.nextLine());
  System.out.print("Enter the accelaration for car B: ");
  a2 = Double.parseDouble(cin.nextLine());

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

  System.out.println("Distance between them: " + Math.abs(s1 - s2) + " meters");

  if (s1 > s2) {
    System.out.println("Car A is first");
  }
  else {
    System.out.println("Car B is first");
  }
}
```

Chapter 18

18.2 Review Questions: True/False

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

18.3 Review Exercises

1. Solution

For input value of 5

Step	Statement	q	b
1	`q = Integer.parseInt(cin.nextLine())`	**5**	?
2	`if (q > 0 && q <= 50)`	true	
3	`b = 1`	5	**1**
4	`System.out.println(b)`	It displays: 1	

For input value of 150

Step	Statement	q	b
1	`q = Integer.parseInt(cin.nextLine())`	**150**	?
2	`if (q > 0 && q <= 50)`	false	
3	`else if (q > 50 && q <= 100)`	false	
4	`else if (q > 100 && q <= 200)`	true	
5	`b = 3`	150	**3**
6	`System.out.println(b)`	It displays: 3	

For input value of 250

Step	Statement	q	b
1	`q = Integer.parseInt(cin.nextLine())`	**250**	?
2	`if (q > 0 && q <= 50)`	false	
3	`else if (q > 50 && q <= 100)`	false	
4	`else if (q > 100 && q <= 200)`	false	
5	`b = 4`	250	**4**
6	`System.out.println(b)`	It displays: 4	

For input value of −1

Step	Statement	q	b
1	`q = Integer.parseInt(cin.nextLine())`	**−1**	?
2	`if (q > 0 && q <= 50)`	false	
3	`else if (q > 50 && q <= 100)`	false	
4	`else if (q > 100 && q <= 200)`	false	

5	`b = 4`	−1	**4**
6	`System.out.println(b)`	It displays: 4	

2. Solution

For input value of 5

Step	Statement	amount	discount	payment
1	`amount = Double.parseDouble(cin.nextLine())`	**5.0**	?	?
2	`discount = 0`	5.0	**0.0**	?
3	`if (amount < 20)`	true		
4	`discount = 0`	5.0	**0.0**	?
5	`payment = amount - amount * discount / 100`	5.0	0.0	**5.0**
6	`System.out.println(discount + ", " + payment)`	It displays: 0.0, 5.0		

For input value of 150

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

For input value of −1

Step	Statement	amount	discount	payment
1	`amount = Double.parseDouble(cin.nextLine())`	**−1.0**	?	?
2	`discount = 0`	−1.0	**0.0**	?
3	`if (amount < 20)`	true		
4	`discount = 0`	−1.0	**0.0**	?
5	`payment = amount - amount * discount / 100`	−1.0	0.0	**−1.0**
6	`System.out.println(discount + ", " + payment)`	It displays: 0.0, −1.0		

3. Solution

```
public static void main(String[] args) {
  double a, y;

  a = Double.parseDouble(cin.nextLine());
```

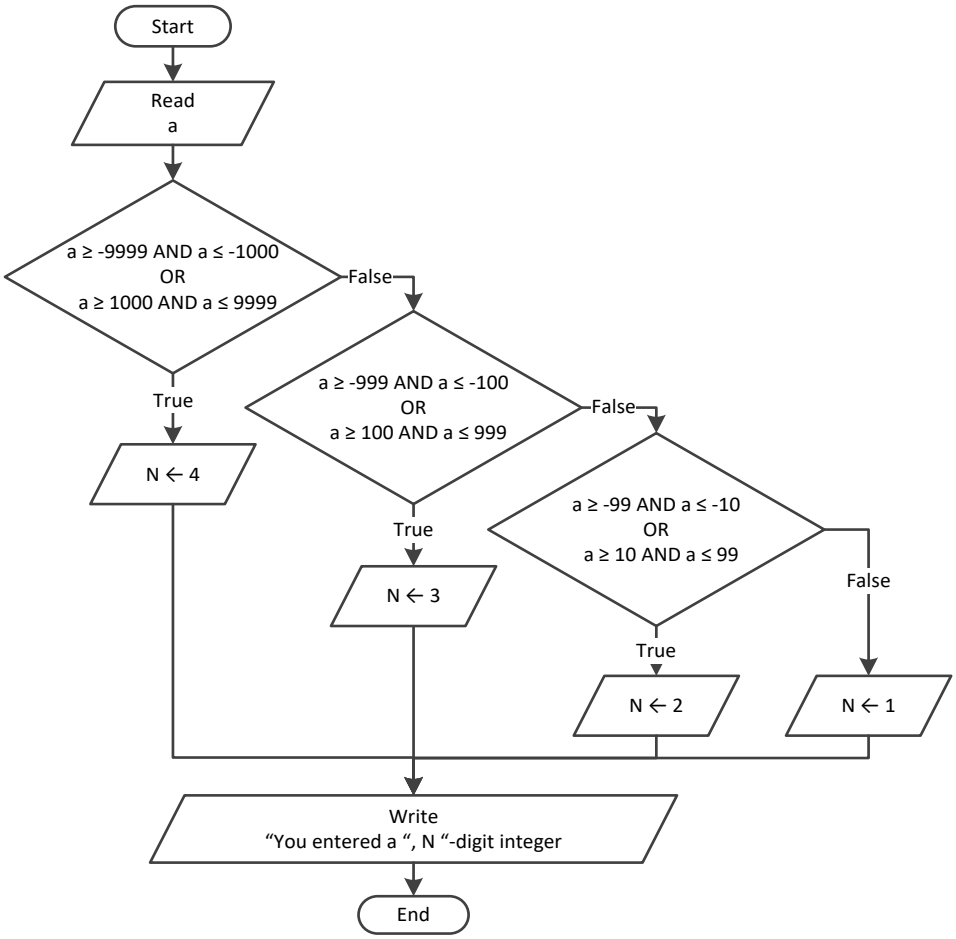
```
  if (a < 1) {
    y = 5 + a;
    System.out.println(y);
  }
  else if (a < 5) {
    y = 23 / a;
    System.out.println(y);
  }
  else if (a < 10) {
    y = 5 * a;
    System.out.println(y);
  }
  else {
    System.out.println("Error!");
  }
}
```

4. Solution

```
public static void main(String[] args) {
  String name1, name2;
  int goals1, goals2;

  System.out.print("Enter team name 1: ");
  name1 = cin.nextLine();
  System.out.print("Enter team name 2: ");
  name2 = cin.nextLine();

  System.out.print("Enter goals " + name1 + " scored: ");
  goals1 = Integer.parseInt(cin.nextLine());
  System.out.print("Enter goals " + name2 + " scored: ");
  goals2 = Integer.parseInt(cin.nextLine());

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

5. Solution

First approach




```
public static void main(String[] args) {
  int a, n;

  a = Integer.parseInt(cin.nextLine());

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

  System.out.println("You entered a " + n + "-digit integer");
}
```

Second approach

```
public static void main(String[] args) {
  int a;
  String a_string;

  a = Integer.parseInt(cin.nextLine());

  a_string = "" + Math.abs(a);
  System.out.println("You entered a " + a_string.length() + "-digit integer");
}
```

6. Solution

First approach

```
public static void main(String[] args) {
  int a, n;

  a = Integer.parseInt(cin.nextLine());

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

Second approach

```
public static void main(String[] args) {
  int a;
  String a_string;

  a = Integer.parseInt(cin.nextLine());

  if (a >= -9999 && a <= 9999) {
    a_string = "" + Math.abs(a);
    System.out.println("You entered a " + a_string.length() + "-digit integer");
  }
  else {
    System.out.println("Error: Invalid value!");
  }
}
```

7. Solution

```
public static void main(String[] args) {
  double cad, eur, gbp, jpy, usd;
  int ch;

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

  System.out.print("Enter a choice: ");
  ch = Integer.parseInt(cin.nextLine());

  System.out.print("Enter an amount in US dollars: ");
  usd = Double.parseDouble(cin.nextLine());

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

8. Solution

```
public static void main(String[] args) {
  int m;

  System.out.print("Enter the number of a month between 1 and 12: ");
  m = Integer.parseInt(cin.nextLine());

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

```
  }
  else {
    System.out.println("Fall (Autumn)");
  }
}
```

9. Solution

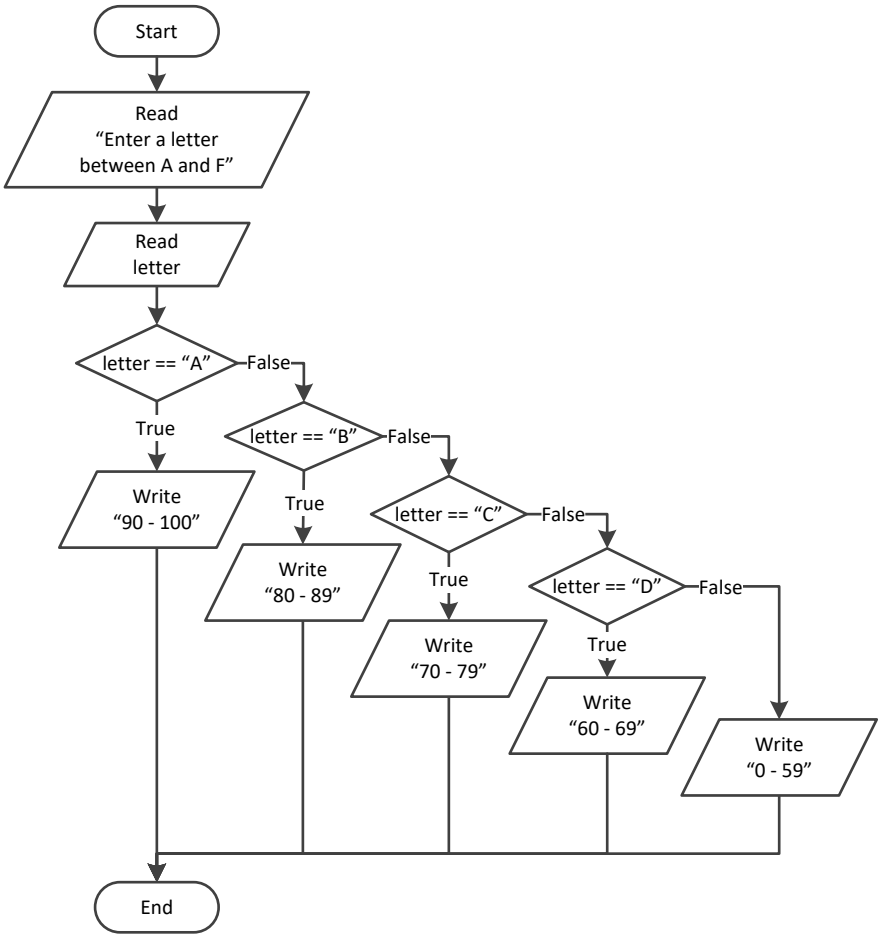
```
public static void main(String[] args) {
  int m;

  System.out.print("Enter the number of a month between 1 and 12: ");
  m = Integer.parseInt(cin.nextLine());

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

10. Solution

```
public static void main(String[] args) {
  double n;
  int x, y;

  System.out.print("Enter a number between 1.0 and 4.9: ");
  n = Double.parseDouble(cin.nextLine());

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

  if (x == 1) {
    System.out.print("One");
  }
  else if (x == 2) {
    System.out.print("Two");
  }
  else if (x == 3) {
    System.out.print("Three");
  }
```

```
  else if (x == 4) {
    System.out.print("Four");
  }

  System.out.print(" point ");

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

11. Solution




```
public static void main(String[] args) {
  String letter;

  System.out.print("Enter a letter between A and F: ");
  letter = cin.nextLine();

  if (letter.equals("A") == true) {
    System.out.println("90 - 100");
  }
  else if (letter.equals("B") == true) {
    System.out.println("80 - 89");
  }
  else if (letter.equals("C") == true) {
    System.out.println("70 - 79");
  }
  else if (letter.equals("D") == true) {
    System.out.println("60 - 69");
  }
  else {
    System.out.println("0 - 59");
  }
}
```

Chapter 19

19.2 Review Questions: True/False

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

19.3 Review Exercises

1. Solution

For input value of 1

Step	Statement	a	x	y
1	`a = Integer.parseInt(cin.nextLine())`	**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	`System.out.println(x + ", " + y)`	It displays: 5, 5		

For input value of 3

Step	Statement	a	x	y
1	`a = Integer.parseInt(cin.nextLine())`	**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	`System.out.println(x + ", " + y)`	It displays: –9, 3		

For input value of 250

Step	Statement	a	x	y
1	`a = Integer.parseInt(cin.nextLine())`	**250**	?	?
2	`x = 0`	250	**0**	?
3	`y = 0`	250	0	**0**
4	`case a == 1`	false		
5	`case a == 2`	false		
6	`case a == 3`	false		

7	`x = x + 3`	250	**3**	0
8	`y++`	250	3	**1**
9	`System.out.println(x + ", " + y)`	It displays: 3, 1		

2. Solution

For input values of 10, 2, 5

Step	Statement	a	x	y
1	`a = Integer.parseInt(cin.nextLine())`	**10**	?	?
2	`x = Integer.parseInt(cin.nextLine())`	10	**2**	?
3	`y = Double.parseDouble(cin.nextLine())`	10	2	**5.0**
4	`case a == 10`	true		
5	`x = x % 2`	10	**0**	5.0
6	`y = Math.pow(y, 2)`	10	0	**25.0**
7	`System.out.println(x + ", " + y)`	It displays: 0, 25.0		

For input values of 5, 2, 3

Step	Statement	a	x	y
1	`a = Integer.parseInt(cin.nextLine())`	**5**	?	?
2	`x = Integer.parseInt(cin.nextLine())`	5	**2**	?
3	`y = Double.parseDouble(cin.nextLine())`	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	`System.out.println(x + ", " + y)`	It displays: 6, 10.0		

For input values of 4, 6, 2

Step	Statement	a	x	y
1	`a = Integer.parseInt(cin.nextLine())`	**4**	?	?
2	`x = Integer.parseInt(cin.nextLine())`	4	**6**	?
3	`y = Double.parseDouble(cin.nextLine())`	4	6	**2.0**
4	`case a == 10`	false		
5	`case a == 3`	false		
6	`case a == 5`	false		
7	`x -= 3`	4	**3**	2.0
8	`y++`	4	3	**3.0**
9	`System.out.println(x + ", " + y)`	It displays: 3, 3.0		

3. Solution

```
public static void main(String[] args) {
  String name;

  System.out.print("Enter the name of a month: ");
  name = cin.nextLine();

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

4. Solution

```
public static void main(String[] args) {
  int choice;
  double feet, inches, miles, yards;

  System.out.println("1. Convert Miles to Yards");
  System.out.println("2. Convert Miles to Feet");
  System.out.println("3. Convert Miles to Inches");

  System.out.print("Enter a choice: ");
  choice = Integer.parseInt(cin.nextLine());

  switch (choice) {
    case 1:
      System.out.print("Enter miles: ");
      miles = Double.parseDouble(cin.nextLine());
      yards = miles * 1760;
      System.out.println(miles + " miles = " + yards + " yards");
      break;
    case 2:
      System.out.print("Enter miles: ");
      miles = Double.parseDouble(cin.nextLine());
      feet = miles * 5280;
      System.out.println(miles + " miles = " + feet + " feet");
      break;
    case 3:
      System.out.print("Enter miles: ");
      miles = Double.parseDouble(cin.nextLine());
      inches = miles * 63360;
      System.out.println(miles + " miles = " + inches + " inches");
      break;
    default:
      System.out.println("Invalid choice!");
  }
}
```

5. Solution

```
public static void main(String[] args) {
  String roman;

  System.out.print("Enter a Roman numeral between I and X: ");
  roman = cin.nextLine();

  switch (roman) {
    case "I":
      System.out.println("1");
      break;
    case "II":
      System.out.println("2");
```

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

6. Solution

```
public static void main(String[] args) {
  int total;

  System.out.print("Enter the total number of CDs purchased in a month: ");
  total = Integer.parseInt(cin.nextLine());

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

```
}
```

7. Solution

```
public static void main(String[] args) {
  int i;
  String name;

  System.out.print("Enter your name: ");
  name = cin.nextLine();

  i = (int)(Math.random() * 3);

  switch (i) {
    case 0:
      System.out.println("Good morning " + name);
      break;
    case 1:
      System.out.println("Good evening " + name);
      break;
    case 2:
      System.out.println("Good night " + name);
      break;
  }
}
```

8. Solution

```
public static void main(String[] args) {
  String num;

  num = cin.nextLine();

  switch (num) {
    case "zero":
      System.out.println(0);
      break;
    case "one":
      System.out.println(1);
      break;
    case "two":
      System.out.println(2);
      break;
    case "three":
      System.out.println(3);
      break;
    case "four":
      System.out.println(4);
      break;
    case "five":
      System.out.println(5);
```

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

9. Solution

```
public static void main(String[] args) {
  int b;

  System.out.print("Enter Beaufort number: ");
  b = Integer.parseInt(cin.nextLine());

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

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

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 = Integer.parseInt(cin.nextLine())`	**20**	?
2	`y = Integer.parseInt(cin.nextLine())`	20	**1**
3	`if (x < 30)`	true	
4	`case y == 1`	true	
5	`x = x % 3`	**2**	1
6	`y = 5`	2	**5**
7	`System.out.println(x + ", " + y)`	It displays: 2, 5	

For input values of 20, 3

Step	Statement	x	y
1	`x = Integer.parseInt(cin.nextLine())`	**20**	?
2	`y = Integer.parseInt(cin.nextLine())`	20	**3**
3	`if (x < 30)`	true	
4	`case y == 1`	false	
5	`case y == 2`	false	
6	`case y == 3`	true	
7	`x = x + 5`	**25**	3
8	`y += 3`	25	**6**
9	`System.out.println(x + ", " + y)`	It displays: 25, 6	

For input values of 12, 8

Step	Statement	x	y
1	`x = Integer.parseInt(cin.nextLine())`	**12**	?
2	`y = Integer.parseInt(cin.nextLine())`	12	**8**
3	`if (x < 30)`	true	
4	`case y == 1`	false	
5	`case y == 2`	false	
6	`case y == 3`	false	
7	`x -= 2`	**10**	8

8	`y++`	10	**9**
9	`System.out.println(x + ", " + y)`	It displays: 10, 9	

For input values of 50, 0

Step	Statement	x	y
1	`x = Integer.parseInt(cin.nextLine())`	**50**	?
2	`y = Integer.parseInt(cin.nextLine())`	50	**0**
3	`if (x < 30)`	false	
4	`y++`	50	**1**
5	`System.out.println(x + ", " + y)`	It displays: 50, 1	

2. Solution

For input values of 60, 25

Step	Statement	x	y
1	`x = Integer.parseInt(cin.nextLine())`	**60**	?
2	`y = Integer.parseInt(cin.nextLine())`	60	25
3	`if ((x + y) / 2 <= 20)`	false	
4	`if (y < 15)`	false	
5	`else if (y < 23)`	false	
6	`x = 2 * x + 5`	**125**	25
7	`y += 1`	125	**26**
8	`System.out.println(x + ", " + y)`	It displays: 125, 26	

For input values of 50, 8

Step	Statement	x	y
1	`x = Integer.parseInt(cin.nextLine())`	**50**	?
2	`y = Integer.parseInt(cin.nextLine())`	50	8
3	`if ((x + y) / 2 <= 20)`	false	
4	`if (y < 15)`	true	
5	`x = x % 4`	**2**	8
6	`y = 2`	2	**2**
7	`System.out.println(x + ", " + y)`	It displays: 2, 2	

For input values of 20, 15

Step	Statement	x	y
1	`x = Integer.parseInt(cin.nextLine())`	**20**	?
2	`y = Integer.parseInt(cin.nextLine())`	20	15
3	`if ((x + y) / 2 <= 20)`	true	
4	`if (y < 10)`	false	
5	`else if (y < 20)`	true	

6	`x = x * 5`	**100**	15
7	`y += 2`	100	**17**
8	`System.out.println(x + ", " + y)`	It displays: 100, 17	

For input values of 10, 30

Step	Statement	x	y
1	`x = Integer.parseInt(cin.nextLine())`	**10**	?
2	`y = Integer.parseInt(cin.nextLine())`	10	30
3	`if ((x + y) / 2 <= 20)`	true	
4	`if (y < 10)`	false	
5	`else if (y < 20)`	false	
6	`x = x - 2`	**8**	30
7	`y += 3`	8	**33**
8	`System.out.println(x + ", " + y)`	It displays: 8, 33	

3. Solution

```
public static void main(String[] args) {
  int a;

  a = Integer.parseInt(cin.nextLine());

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

4. Solution

```
public static void main(String[] args) {
  double a, b, c;

  System.out.print("Enter the three sides of a triangle: ");
  a = Double.parseDouble(cin.nextLine());
  b = Double.parseDouble(cin.nextLine());
```

```
  c = Double.parseDouble(cin.nextLine());

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

      System.out.println("Right (or right-angled)");
    }
    else {
      System.out.println("Not special");
    }
  }
}
```

5. Solution

```
public static void main(String[] args) {
  int amount, pin, r, usd1, usd10, usd5;

  System.out.print("Enter your four-digit PIN : ");
  pin= Integer.parseInt(cin.nextLine());
  if (pin != 1234) {
    System.out.print("Wrong PIN. Enter your four-digit PIN : ");
    pin = Integer.parseInt(cin.nextLine());
    if (pin != 1234) {
      System.out.print("Wrong PIN. Enter your four-digit PIN : ");
      pin = Integer.parseInt(cin.nextLine());
    }
  }

  if (pin != 1234) {
    System.out.println("PIN locked!");
  }
  else {
    System.out.print("Enter the amount of money (an integer value) that you want to withdraw: ");
    amount = Integer.parseInt(cin.nextLine());
    usd10 = (int)(amount / 10);
    r = amount % 10;
    usd5 = (int)(r / 5);
    usd1 = r % 5;
    System.out.print(usd10 + " note(s) of $10 " + usd5 + " note(s) of $5 ");
    System.out.println("and " + usd1 + " note(s) of $1");
  }
}
```

6. Solution

First approach

```
public static void main(String[] args) {
  double t, w;

  System.out.print("Enter temperature (in Fahrenheit): ");
  t = Double.parseDouble(cin.nextLine());
  System.out.print("Enter wind speed (in miles/hour): ");
  w = Double.parseDouble(cin.nextLine());

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

Second approach

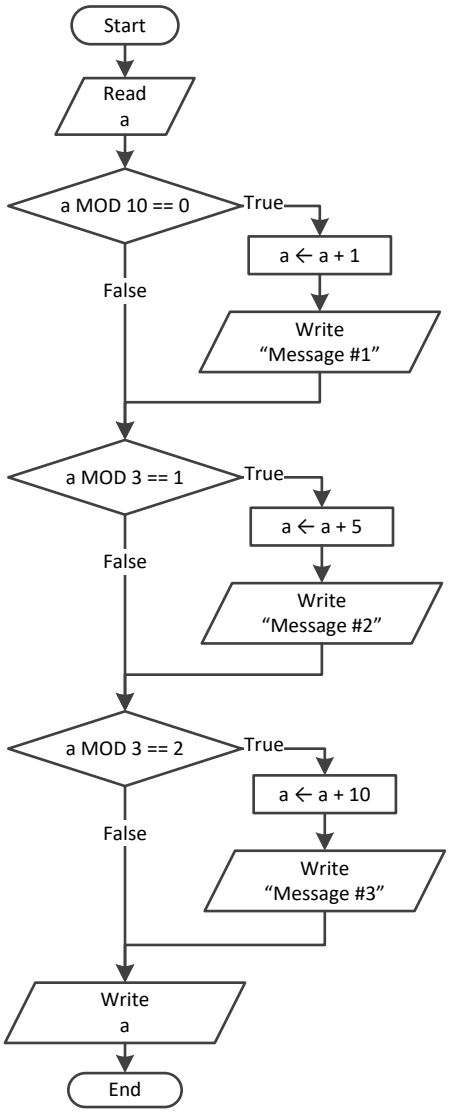
```
public static void main(String[] args) {
  double t, w;
  String message1, message2;

  System.out.print("Enter temperature (in Fahrenheit): ");
  t = Double.parseDouble(cin.nextLine());
  System.out.print("Enter wind speed (in miles/hour): ");
  w = Double.parseDouble(cin.nextLine());

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

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

```
  }

  System.out.println("The day is " + message1 + " and " + message2);
}
```

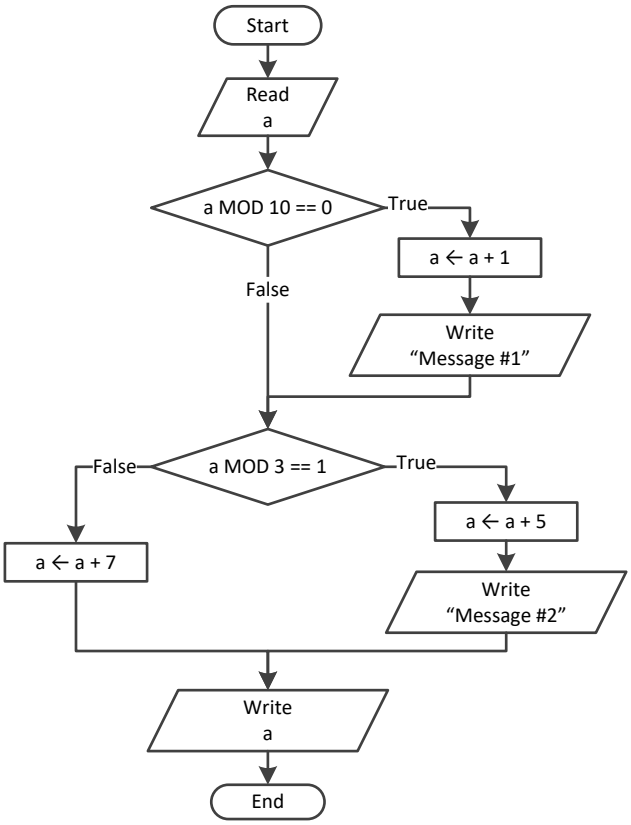
Chapter 21

21.4 Review Exercises

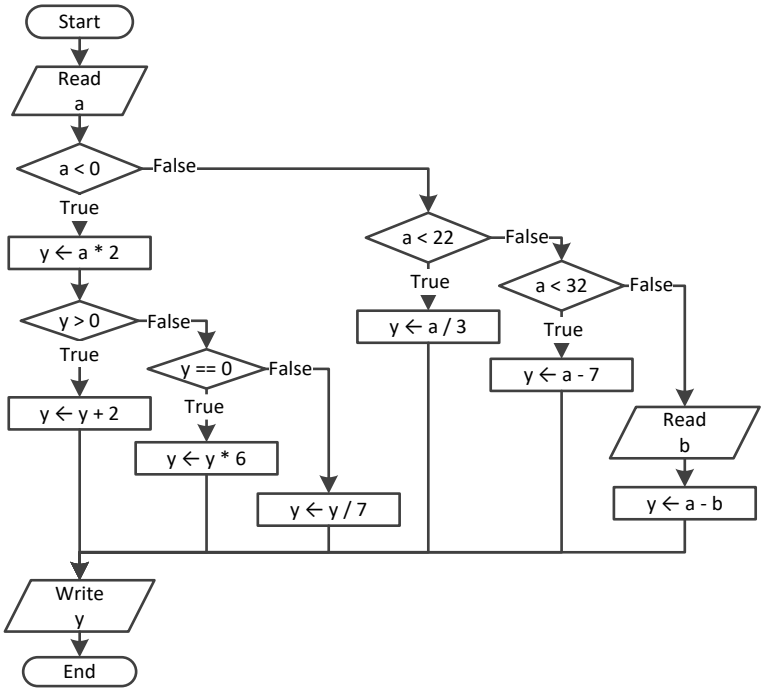
1. Solution



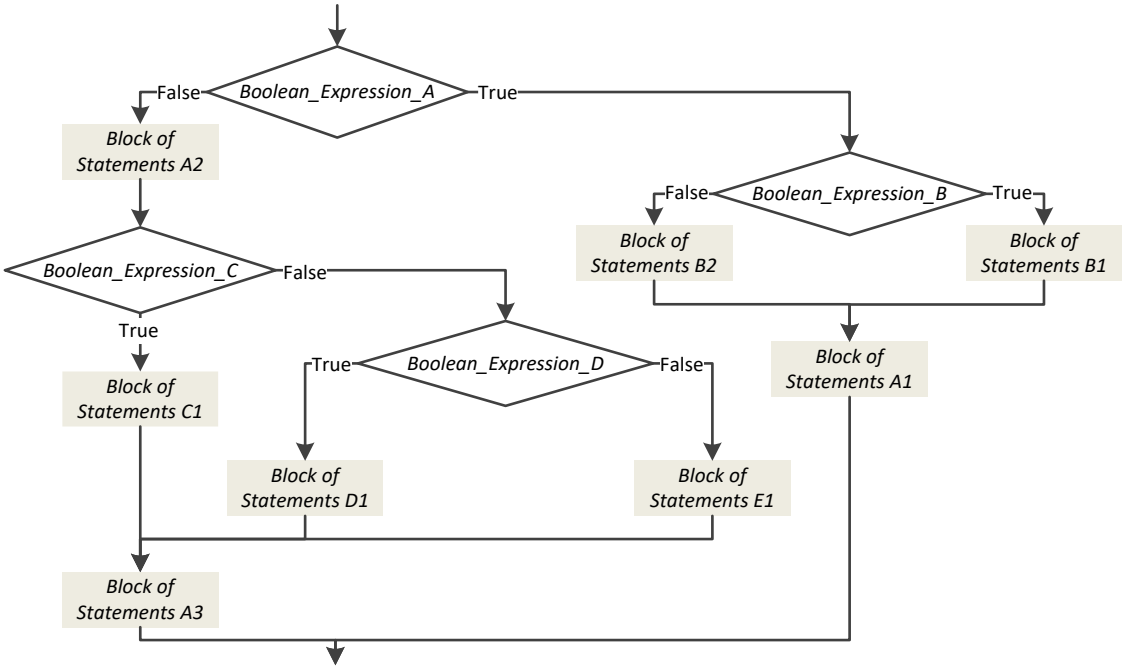
2. Solution



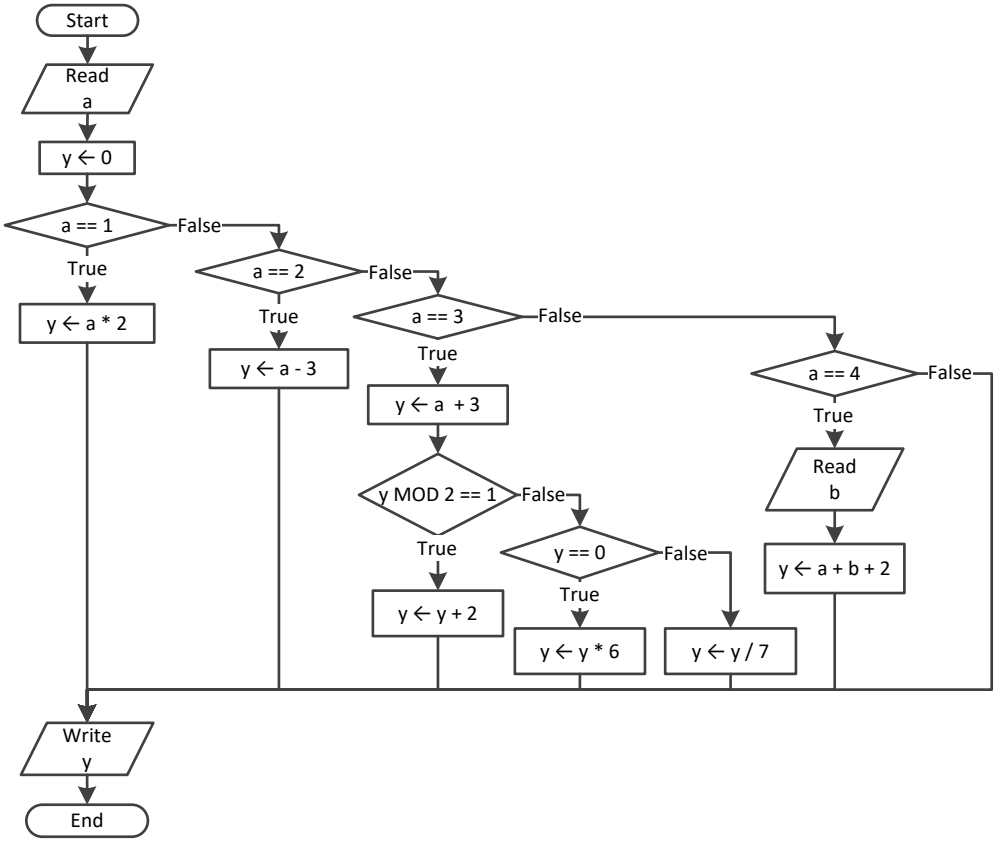
3. Solution



4. Solution



5. Solution



6. Solution

```
public static void main(String[] args) {
  double x, y, z;

  x = Double.parseDouble(cin.nextLine());
  y = Double.parseDouble(cin.nextLine());

  if (x != 100 || y <= 10) {
    z = Double.parseDouble(cin.nextLine());
    if (z <= x + y) {
      x -= 3;
      y = x + 4;
    }
  }
  System.out.println(x + " " + y);
}
```

7. Solution

```
public static void main(String[] args) {
  int x;

  x = Integer.parseInt(cin.nextLine());

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

8. Solution

```
public static void main(String[] args) {
  int a, b, c, d, y;

  a = Integer.parseInt(cin.nextLine());
  b = Integer.parseInt(cin.nextLine());
```

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

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

  System.out.println(y);
}
```

9. Solution

```
public static void main(String[] args) {
  int x;

  x = Integer.parseInt(cin.nextLine());

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

10. Solution

```
public static void main(String[] args) {
```

```
  double a, b, y;

  a = Double.parseDouble(cin.nextLine());
  b = Double.parseDouble(cin.nextLine());

  y = a * b;

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

11. Solution

```
public static void main(String[] args) {
  double a, b, c;

  a = Double.parseDouble(cin.nextLine());
  b = Double.parseDouble(cin.nextLine());
  c = Double.parseDouble(cin.nextLine());

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

```
  }
  System.out.println(a + " " + b  + " " + c);
}
```

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

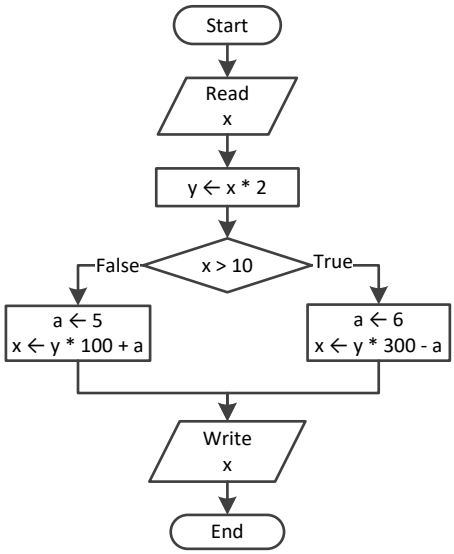
1. a
2. b
3. a
4. c

22.11 Review Exercises

1. Solution

```
public static void main(String[] args) {
  int a, x, y;

  y = Integer.parseInt(cin.nextLine());
  x = Integer.parseInt(cin.nextLine());

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

2. Solution



3. Solution

```
public static void main(String[] args) {
  double a, y;

  a = Double.parseDouble(cin.nextLine());

  if (a >= 10) {
    System.out.println("Error!");
  }
  else {
    if (a < 1) {
      y = 5 + a;
    }
    else if (a < 5) {
      y = 23 / a;
    }
    else {
      y = 5 * a;
    }
    System.out.println(y);
  }
}
```

4. Solution

```
public static void main(String[] args) {
  int day, month;
  String name;

  day = Integer.parseInt(cin.nextLine());
  month = Integer.parseInt(cin.nextLine());
  name = cin.nextLine();

  if (day == 16 && month == 2 && name.equals("Loukia") == true) {
    System.out.println("Happy Birthday!!!");
  }
  else {
    System.out.println("No match!");
  }
}
```

5. Solution

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

```
public static void main(String[] args) {
  double a, b, c, d;

  a = Double.parseDouble(cin.nextLine());
  b = Double.parseDouble(cin.nextLine());
  c = Double.parseDouble(cin.nextLine());
```

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

6. Solution

```
public static void main(String[] args) {
  double a, b, c, d;

  a = Double.parseDouble(cin.nextLine());
  b = Double.parseDouble(cin.nextLine());
  c = Double.parseDouble(cin.nextLine());

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

  if (a <= 10) {
    System.out.println("Error!");
  }
}
```

7. Solution

```
public static void main(String[] args) {
  int a, b, y;

  a = Integer.parseInt(cin.nextLine());
  b = Integer.parseInt(cin.nextLine());

  y = 3;
  if (a > 0) {
    y = y * a;
    System.out.println("Hello Zeus");
  }

  System.out.println(y + " " + b);
}
```

8. Solution

```
public static void main(String[] args) {
  double a, b, y;

  a = Double.parseDouble(cin.nextLine());
  b = Double.parseDouble(cin.nextLine());

  y = 0;
  if (a > 0) {
    y = y + 7;
  }
  else {
    System.out.println("Hello Zeus");
    System.out.println(Math.abs(a));
  }
  System.out.println(y);
}
```

9. Solution

```
public static void main(String[] args) {
  String os;

  System.out.print("What is your tablet's OS? ");
  os = cin.nextLine();

  if (os.equals("iOS") == true) {
    System.out.println("Apple");
  }
  else if (os.equals("Android") == true) {
    System.out.println("Google");
  }
  else if (os.equals("Windows") == true) {
    System.out.println("Microsoft");
  }
}
```

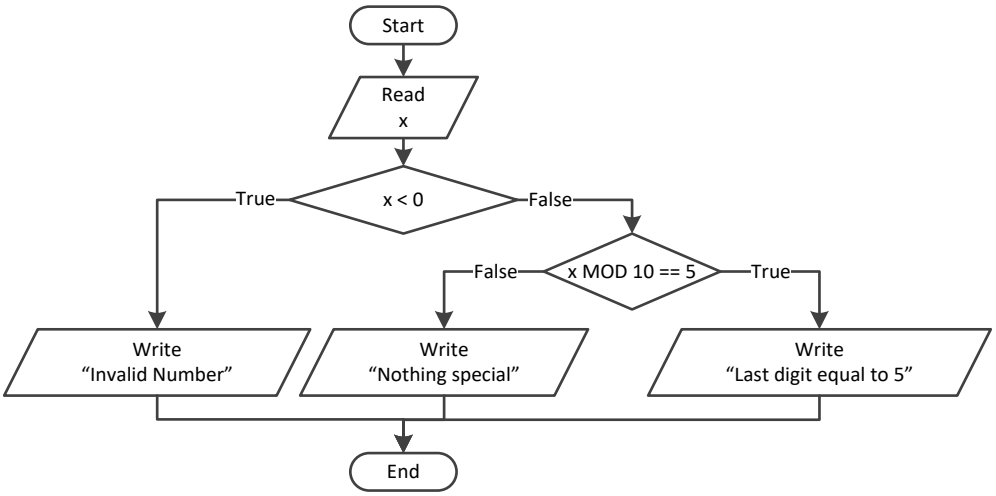
Chapter 23

23.6 Review Exercises

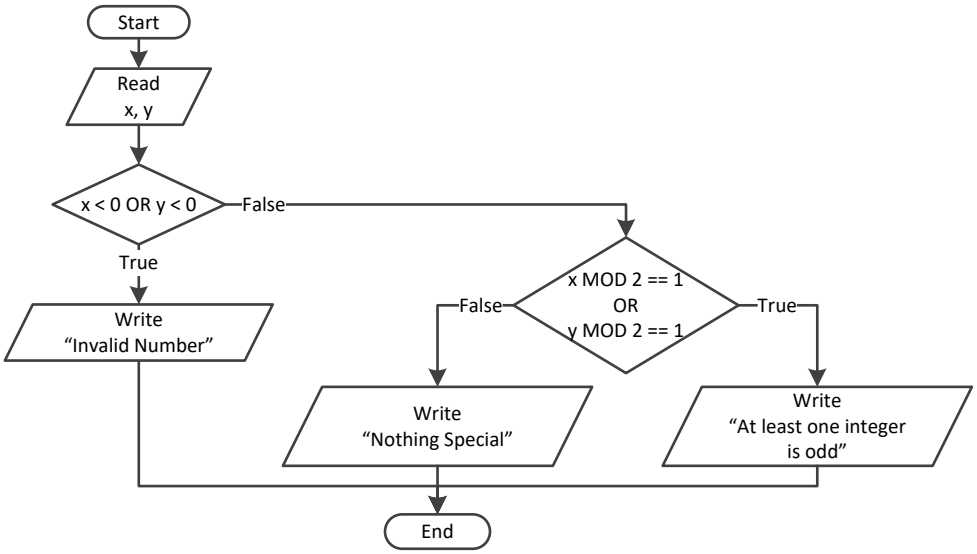
1. Solution

```
public static void main(String[] args) {
  double x;

  System.out.print("Enter a non-negative number: ");
  x = Double.parseDouble(cin.nextLine());
  if (x < 0) {
    System.out.println("Error! You entered a negative value");
  }
  else {
    System.out.println("The square root of " + x + " is " + Math.sqrt(x));
  }
}
```

2. Solution

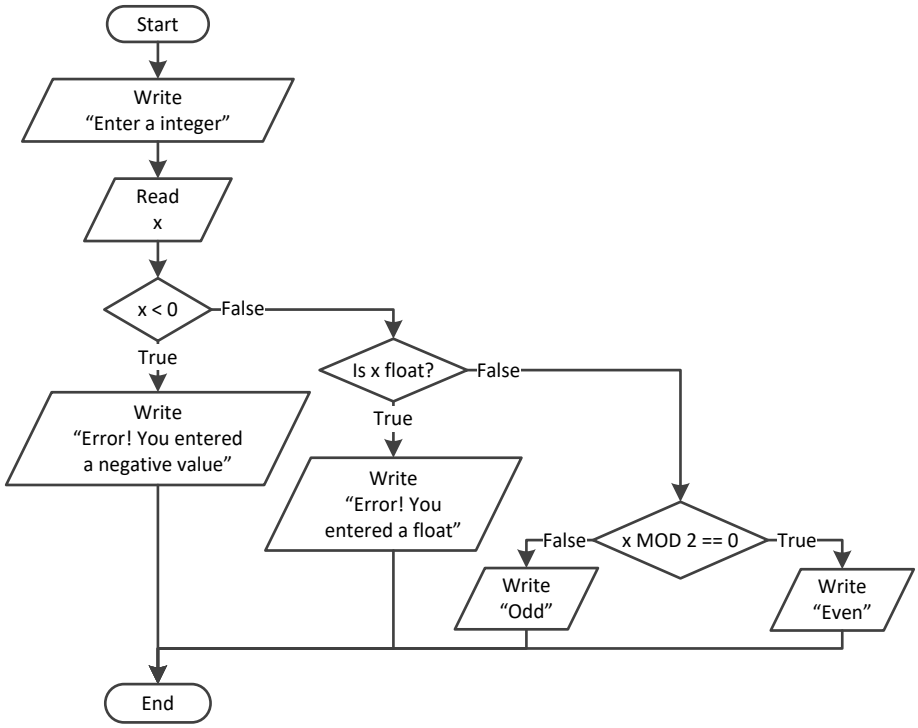


3. Solution



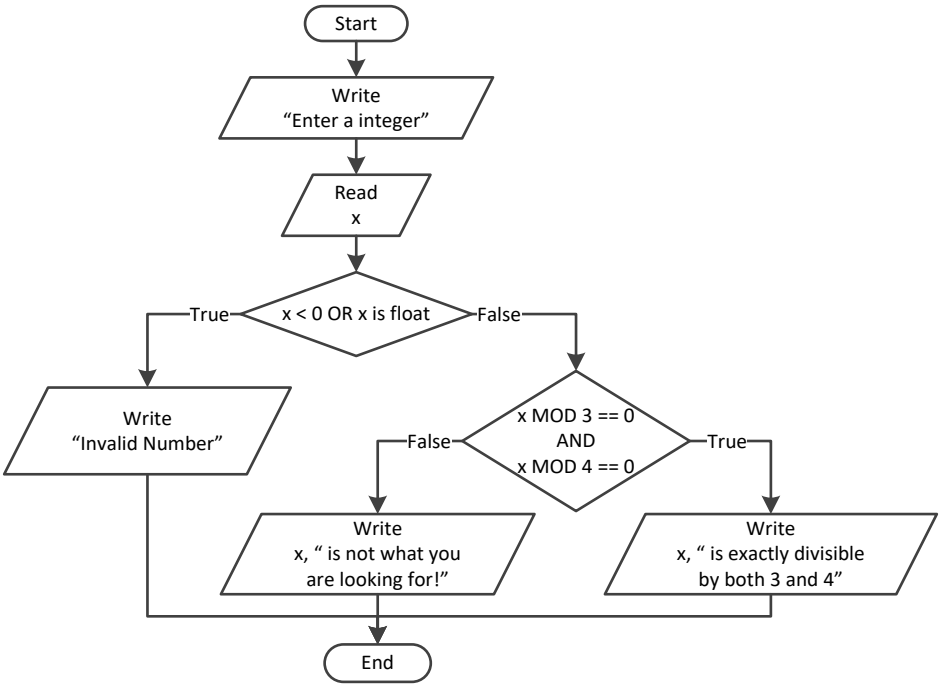

```
public static void main(String[] args) {
  int x, y;

  x = Integer.parseInt(cin.nextLine());
  y = Integer.parseInt(cin.nextLine());

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

4. Solution



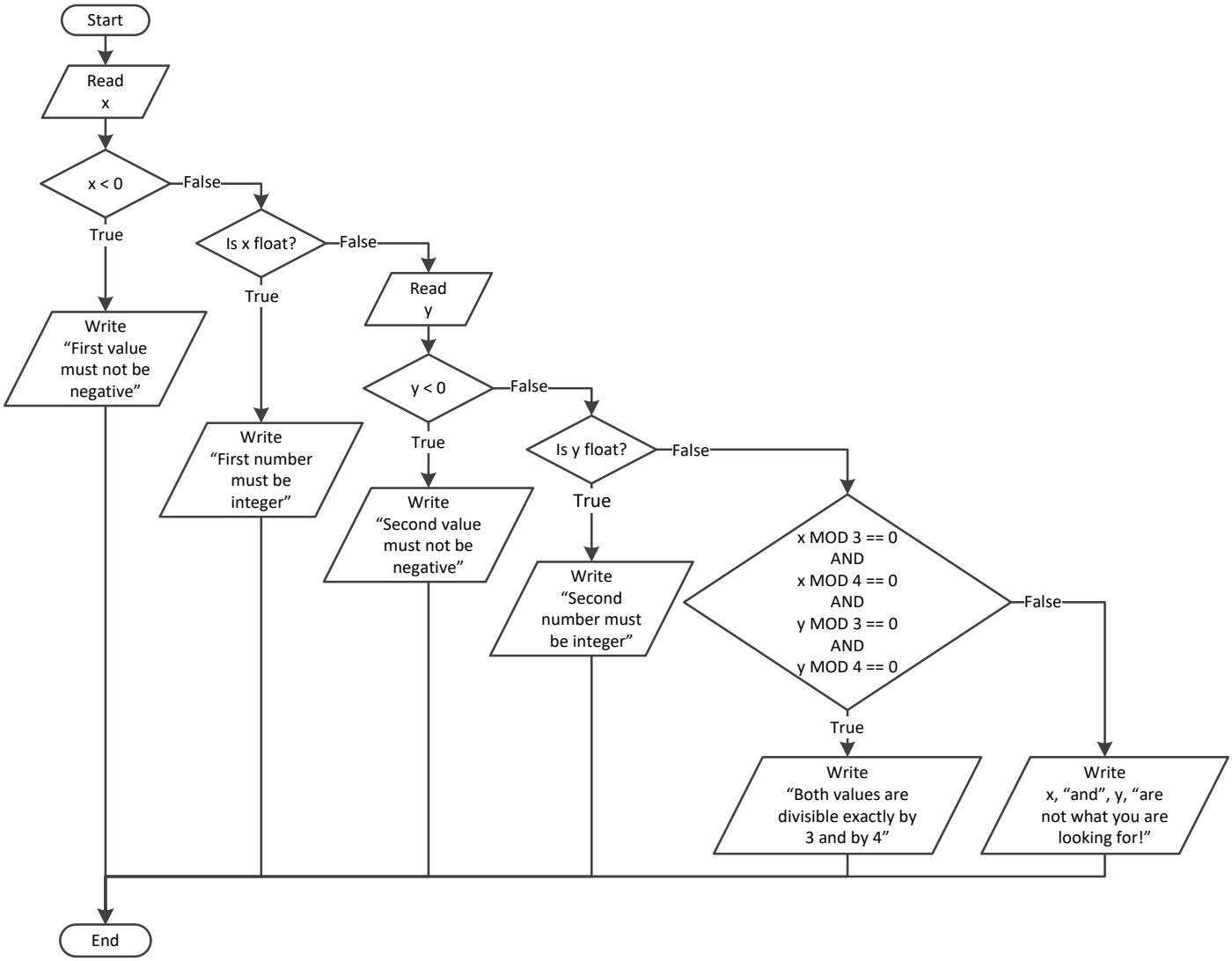

```
public static void main(String[] args) {
  double x;

  System.out.print("Enter a non-negative number: ");
  x = Double.parseDouble(cin.nextLine());
  if (x < 0) {
    System.out.println("Error! You entered a negative value");
  }
  else if (x != (int)x) {
    System.out.println("Error! You entered a float");
  }
  else if (x % 2 == 0) {
    System.out.println("Even");
  }
  else {
    System.out.println("Odd");
  }
}
```

5. Solution




```
public static void main(String[] args) {
  double x;

  System.out.print("Enter an integer: ");
  x = Double.parseDouble(cin.nextLine());

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

6. Solution




```
public static void main(String[] args) {
  double x, y;

  x = Double.parseDouble(cin.nextLine());

  if (x < 0) {
    System.out.println("First value must be not be negative");
  }
  else {
    if (x != (int)x) {
        System.out.println("First number must be integer");
    }
    else {
      y = Double.parseDouble(cin.nextLine());
      if (y < 0) {
        System.out.println("Second value must be not be negative");
      }
```

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

7. Solution

```
public static void main(String[] args) {
  int choice;
  double t;

  System.out.println("1. Convert Kelvin to Fahrenheit");
  System.out.println("2. Convert Fahrenheit to Kelvin");
  System.out.println("3. Convert Fahrenheit to Celsius");
  System.out.println("4. Convert Celsius to Fahrenheit");

  System.out.print("Enter a choice: ");
  choice = Integer.parseInt(cin.nextLine());
  System.out.print("Enter a temperature: ");
  t = Double.parseDouble(cin.nextLine());

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

```
          System.out.println((t + 459.57) / 1.8);
        }
        break;
      case 3:
        if (t < -459.67) {  //Absolute zero in Fahrenheit
          System.out.println("Wrong temperature");
        }
        else {
          System.out.println(5 / 9 * (t - 32));
        }
        break;
      case 4:
        if (t < -273.15) {  //Absolute zero in Celcius
          System.out.println("Wrong temperature");
        }
        else {
          System.out.println(9 / 5 * t + 32);
        }
        break;
    }
  }
}
```

8. Solution

```
public static void main(String[] args) {
  int a, b;
  String op;

  System.out.print("Enter 1st integer: ");
  a = Integer.parseInt(cin.nextLine());
  System.out.print("Enter type of operation: ");
  op = cin.nextLine();
  System.out.print("Enter 2nd integer: ");
  b = Integer.parseInt(cin.nextLine());

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

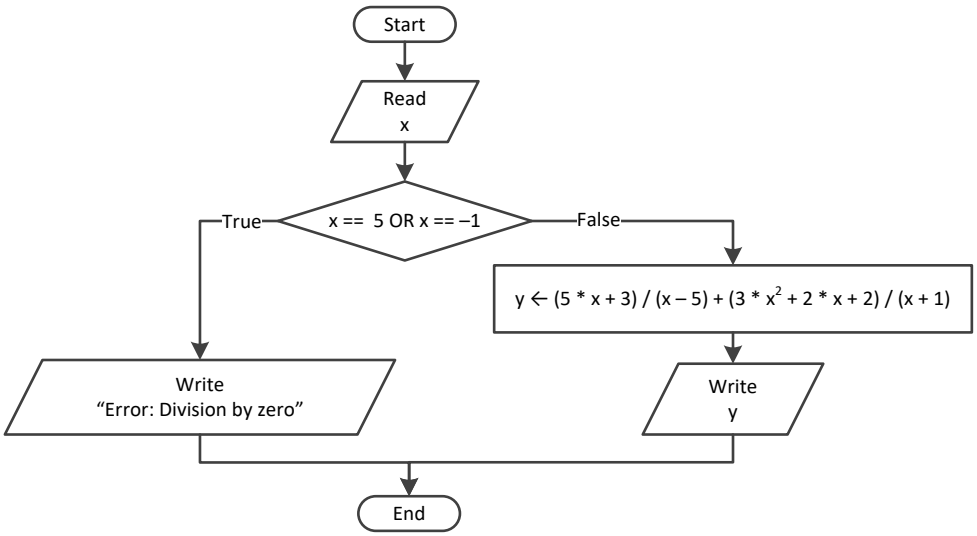
```
        System.out.println(a / (double)b);
      }
      break;
    case "DIV":
      if (b == 0) {
        System.out.println("Error: Division by zero");
      }
      else {
        System.out.println((int)(a / b));
      }
      break;
    case "MOD":
      if (b == 0) {
        System.out.println("Error: Division by zero");
      }
      else {
        System.out.println(a % b);
      }
      break;
    case "POWER":
      System.out.println(Math.pow(a, b));
      break;
  }
}
```

9. Solution

```
public static void main(String[] args) {
  int a, b;
  String op;

  System.out.print("Enter 1st integer: ");
  a = Integer.parseInt(cin.nextLine());
  System.out.print("Enter type of operation: ");
  op = cin.nextLine();
  System.out.print("Enter 2nd integer: ");
  b = Integer.parseInt(cin.nextLine());

  switch (op) {
    case "+":
      System.out.println(a + b);
      break;
    case "-":
      System.out.println(a - b);
      break;
    case "*":
      System.out.println(a * b);
      break;
    case "/":
      if (b == 0) {
        System.out.println("Error: Division by zero");
```

```
      }
      else {
        System.out.println(a / (double)b);
      }
      break;
    case "DIV":
      if (b == 0) {
        System.out.println("Error: Division by zero");
      }
      else {
        System.out.println((int)(a / b));
      }
      break;
    case "MOD":
      if (b == 0) {
        System.out.println("Error: Division by zero");
      }
      else {
        System.out.println(a % b);
      }
      break;
    case "POWER":
      System.out.println(Math.pow(a, b));
      break;
    default:
      System.out.println("Error: Invalid operator");
  }
}
```

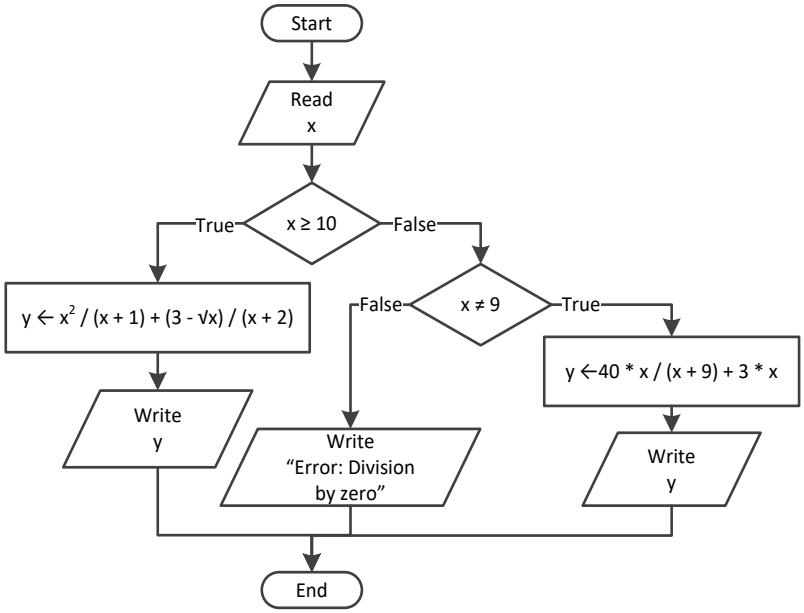
10. Solution




```
public static void main(String[] args) {
  double x, y;

  x = Double.parseDouble(cin.nextLine());
```

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

11. Solution




```
public static void main(String[] args) {
  double x, y;

  x = Double.parseDouble(cin.nextLine());
  if (x >= 10) {
    y = Math.pow(x, 2) / (x + 1) + (3 - Math.sqrt(x)) / (x + 2);
    System.out.println(y);
  }
  else if (x != 9) {
      y = 40 * x / (x + 9) + 3 * x;
      System.out.println(y);
  }
  else {
    System.out.println("Error: Division by zero");
  }
}
```

12. Solution

```
public static void main(String[] args) {
  double x, y;

  x = Double.parseDouble(cin.nextLine());
```

```
  if (x <= -15 || x > 25) {
    y = x - 1;
    System.out.println(y);
  }
  else if (x <= -10) {
    y = x / Math.sqrt(x + 30) + Math.pow(8 + x, 2) / (x + 1);
    System.out.println(y);
  }
  else if (x <= 0) {
    y = Math.abs(40 * x) / (x - 8);
    System.out.println(y);
  }
  else {
    if (x == 9) {
      System.out.println("Error: Division by zero");
    }
    else if (x < 9) {
      System.out.println("Error: Invalid square root");
    }
    else {
      y = 3 * x / Math.sqrt(x - 9);
      System.out.println(y);
    }
  }
}
```

13. Solution

```
public static void main(String[] args) {
  int a1, a2, a3, maximum, minimum;
  String max_name, min_name, n1, n2, n3;

  System.out.print("Enter the age of the first person: ");
  a1 = Integer.parseInt(cin.nextLine());
  System.out.print("Enter the name of the first person: ");
  n1 = cin.nextLine();
  System.out.print("Enter the age of the second person: ");
  a2 = Integer.parseInt(cin.nextLine());
  System.out.print("Enter the name of the second person: ");
  n2 = cin.nextLine();
  System.out.print("Enter the age of the third person: ");
  a3 = Integer.parseInt(cin.nextLine());
  System.out.print("Enter the name of the third person: ");
  n3 = cin.nextLine();

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

```
  if (a3 < minimum) {
    minimum = a3;
    min_name = n3;
  }

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

  System.out.println(min_name + " " + max_name);
}
```

14. Solution

```
public static void main(String[] args) {
  int age1, age2, age3, maximum, middle, minimum;

  System.out.print("Enter age for person No1:");
  age1 = Integer.parseInt(cin.nextLine());
  System.out.print("Enter age for person No2:");
  age2 = Integer.parseInt(cin.nextLine());
  System.out.print("Enter age for person No3:");
  age3 = Integer.parseInt(cin.nextLine());

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

  middle = age1 + age2 + age3 - minimum - maximum;
  System.out.println(middle);
}
```

15. Solution

```
public static void main(String[] args) {
  int a1, a2, a3, maximum, minimum, middle;
  String max_name, min_name, n1, n2, n3;

  System.out.print("Enter the age of the first person: ");
  a1 = Integer.parseInt(cin.nextLine());
  System.out.print("Enter the name of the first person: ");
  n1 = cin.nextLine();
  System.out.print("Enter the age of the second person: ");
  a2 = Integer.parseInt(cin.nextLine());
  System.out.print("Enter the name of the second person: ");
  n2 = cin.nextLine();
  System.out.print("Enter the age of the third person: ");
  a3 = Integer.parseInt(cin.nextLine());
  System.out.print("Enter the name of the third person: ");
  n3 = cin.nextLine();

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

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

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

  if (Math.abs(maximum - middle) < Math.abs(minimum - middle)) {
    System.out.println(max_name);
  }
  else {
    System.out.println(min_name);
  }
}
```

16. Solution

```
public static void main(String[] args) {
  int digit1, digit2, digit3, r, total, x;
  double x_dbl;

  System.out.print("Enter a three-digit integer: ");
  x_dbl = Double.parseDouble(cin.nextLine());

  if (x_dbl != (int)x_dbl) {
    System.out.println("Error! You must enter an integer");
  }
  else if (x_dbl < 100 || x_dbl > 999) {
    System.out.println("Entered integer is not a three-digit integer");
  }
  else {
    x = (int)x_dbl;

    digit1 = (int)(x / 100);
    r = (int)x % 100;

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

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

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

17. Solution

```
public static void main(String[] args) {
  int d, m, y;

  System.out.print("Enter day 1 - 31: ");
  d = Integer.parseInt(cin.nextLine());
  System.out.print("Enter month 1 - 12: ");
  m = Integer.parseInt(cin.nextLine());
  System.out.print("Enter year: ");
  y = Integer.parseInt(cin.nextLine());

  if (m == 2) {
    if (y % 4 == 0 && y % 100 != 0 || y % 400 == 0) {
      System.out.println(29 - d);
    }
```

```
    else {
      System.out.println(28 - d);
    }
  }
  else if (m == 4 || m == 6 || m == 9 || m == 11) {
    System.out.println(30 - d);
  }
  else {
    System.out.println(31 - d);
  }
}
```

18. Solution

```
public static void main(String[] args) {
  String word, word1, word2;

  word = cin.nextLine();

  //Using substring() instead of charAt() method is more convenient in this case
  word1 = word.substring(0, 1).toUpperCase() +
          word.substring(1, 2).toLowerCase() +
          word.substring(2, 3).toUpperCase() +
          word.substring(3, 4).toLowerCase() +
          word.substring(4, 5).toUpperCase() +
          word.substring(5, 6).toLowerCase();

  word2 = word.substring(0, 1).toLowerCase() +
          word.substring(1, 2).toUpperCase() +
          word.substring(2, 3).toLowerCase() +
          word.substring(3, 4).toUpperCase() +
          word.substring(4, 5).toLowerCase() +
          word.substring(5, 6).toUpperCase();

  if (word.equals(word1) || word.equals(word2)) {
    System.out.println("Word is okay!");
  }
  else {
    System.out.println("Word is not okay");
  }
}
```

19. Solution

```
public static void main(String[] args) {
  int q;
  double discount, payment;

  System.out.print("Enter quantity: ");
  q = Integer.parseInt(cin.nextLine());
```

```
  if (q < 3) {
    discount = 0;
  }
  else if (q < 6) {
    discount = 10;
  }
  else if (q < 10) {
    discount = 15;
  }
  else if (q < 14) {
    discount = 20;
  }
  else if (q < 20) {
    discount = 27;
  }
  else {
    discount = 30;
  }

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

  System.out.println("You got a discount of " + discount + "%");
  System.out.println("You must pay $" + payment);
}
```

20. Solution

```
static final double VAT = 0.19;

public static void main(String[] args) {
  double amount, discount, payment;

  System.out.print("Enter a before-tax amount: : ");
  amount = Double.parseDouble(cin.nextLine());

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

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

    System.out.println("You got a discount of " + discount + "%");
    System.out.println("You must pay $" + payment);
  }
}
```

21. Solution

```
public static void main(String[] args) {
  int a, h, w;
  double bmi;

  System.out.print("Enter age: ");
  a = Integer.parseInt(cin.nextLine());
  if (a < 18) {
    System.out.println("Invalid age");
  }
  else {
    System.out.print("Enter weight in pounds: ");
    w = Integer.parseInt(cin.nextLine());
    System.out.print("Enter height in inches: ");
    h = Integer.parseInt(cin.nextLine());

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

    if (bmi < 15) {
      System.out.println("Very severely underweight");
    }
    else if (bmi < 16) {
      System.out.println("Severely underweight");
    }
    else if (bmi < 18.5) {
      System.out.println("Underweight");
    }
    else if (bmi < 25) {
      System.out.println("Normal");
    }
    else if (bmi < 30) {
      System.out.println("Overweight");
    }
    else if (bmi < 35) {
      System.out.println("Severely overweight");
    }
    else {
      System.out.println("Very severely overweight");
    }
  }
}
```

22. Solution

```
static final double TAX_RATE = 0.10;

public static void main(String[] args) {
  int water;
  double total;

  System.out.print("Enter water consumption (in cubic feet): ");
  water = Integer.parseInt(cin.nextLine());

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

    total = total + total * TAX_RATE;
    System.out.println("Total amount to pay (taxes included): " + total);
  }
}
```

23. Solution

```
public static void main(String[] args) {
  int children;
  double income, tax;

  System.out.print("Enter taxable income: ");
  income = Double.parseDouble(cin.nextLine());
  System.out.print("Enter number of children: ");
  children = Integer.parseInt(cin.nextLine());

  if (income <= 8000) {
    tax = income * 0.10;
  }
  else if (income <= 30000) {
    tax = 8000 * 0.10 + (income - 8000) * 0.15;
  }
  else if (income <= 70000) {
```

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

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

24. Solution

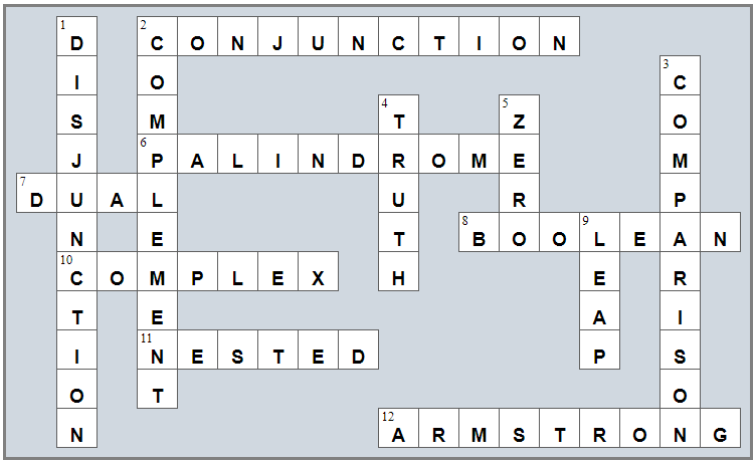
```
public static void main(String[] args) {
  double wind;

  System.out.print("Enter wind speed (in miles/hour): ");
  wind = Double.parseDouble(cin.nextLine());

  if (wind < 0) {
    System.out.println("Error! You entered a negative value");
  }
  else {
    if (wind < 1) {
      System.out.println("Beaufort: 0\nCalm");
    }
    else if (wind < 4) {
      System.out.println("Beaufort: 1\nLight air");
    }
    else if (wind < 8) {
      System.out.println("Beaufort: 2\nLight breeze");
    }
    else if (wind < 13) {
      System.out.println("Beaufort: 3\nGentle breeze");
    }
    else if (wind < 18) {
      System.out.println("Beaufort: 4\nModerate breeze");
    }
    else if (wind < 25) {
      System.out.println("Beaufort: 5\nFresh breeze");
    }
    else if (wind < 31) {
      System.out.println("Beaufort: 6\nStrong breeze");
    }
    else if (wind < 39) {
      System.out.println("Beaufort: 7\nModerate gale");
    }
    else if (wind < 47) {
      System.out.println("Beaufort: 8\nGale");
    }
```

```
    else if (wind < 55) {
      System.out.println("Beaufort: 9\nStrong gale");
    }
    else if (wind < 64) {
      System.out.println("Beaufort: 10\nStorm");
    }
    else if (wind < 74) {
      System.out.println("Beaufort: 11\nViolent storm");
    }
    else {
      System.out.println("Beaufort: 12\nHurricane force");
    }

    if (wind < 13) {
      System.out.println("It's Fishing Day!!!");
    }
  }
}
```

Review in “Decision Control Structures”

Review Crossword Puzzle

1.



Chapter 24

24.3 Review Questions: True/False

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

Chapter 25

25.4 Review Questions: True/False

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

```
public static void main(String[] args) {
  int i;

  i = 3;
  do {
    System.out.println(i);
    i--;
  } while (i > 0);
  System.out.println("The end");
}
```

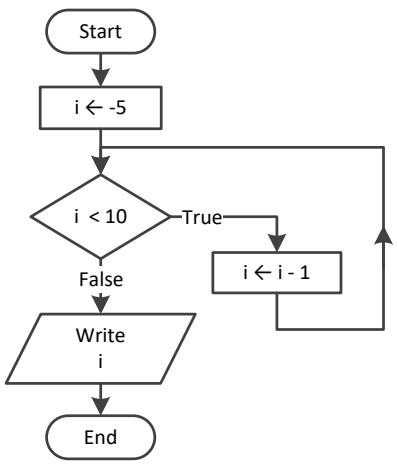
2. Solution

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

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

It performs 4 iterations

3. Solution



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

It performs an infinite number of iterations

4. Solution

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

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

5. Solution

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

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

6. Solution

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

7. Solution

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

18	`while (y < x)`	false

8. Solution

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

9. Solution

Step	Statement	x	y
1	`y = 2`	?	**2**
2	`x = 0`	**0**	2
3	`y = Math.pow (y, 2)`	0	**4**
4	`if (x < 256)`	true	
5	`x = x + y`	**4**	
6	`System.out.println(x + ", " + y)`	It displays: 4, 4	
7	`while (y < 65535)`	true	
8	`y = Math.pow (y, 2)`	4	**16**
9	`if (x < 256)`	true	

10	`x = x + y`	**20**	16
11	`System.out.println(x + ", " + y)`	It displays: 20, 16	
12	`while (y < 65535)`	true	
13	`y = Math.pow (y, 2)`	20	**256**
14	`if (x < 256)`	true	
15	`x = x + y`	**276**	256
16	`System.out.println(x + ", " + y)`	It displays: 276, 256	
17	`while (y < 65535)`	true	
18	`y = Math.pow (y, 2)`	276	**65536**
19	`if (x < 256)`	false	
20	`System.out.println(x + ", " + y)`	It displays: 276, 65536	
21	`while (y < 65535)`	false	

10. Solution

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

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

11. Solution

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

12. Solution

i. 4
ii. −2
iii. 2
iv. 10

13. Solution

```
public static void main(String[] args) {
  double a, total;
  int i, n;

  n = Integer.parseInt(cin.nextLine());
  total = 0

  i = 1;
  while (i <= n) {
    a = Double.parseDouble(cin.nextLine());
    total = total + a;
```

```
    i++;
  }

  System.out.println(total);
  if (n > 0) {
    System.out.println(total / n);
  }
}
```

14. Solution

```
public static void main(String[] args) {
  int a, i, n, p;
  int count = 0;

  n = Integer.parseInt(cin.nextLine());
  p = 1;

  i = 1;
  while (i <= n) {
    a = Integer.parseInt(cin.nextLine());
    if (a % 2 == 0) {
      p = p * a;
      count++;
    }
    i++;
  }

  if (count > 0) {
    System.out.println(p);
  }
  else {
    System.out.println("You entered no even integers ");
  }
}
```

15. Solution

```
public static void main(String[] args) {
  int a, i, total;

  total = 0;

  i = 1;
  while (i <= 100) {
    a = Integer.parseInt(cin.nextLine());
    if (a % 10 == 0) {
      total = total + a;
    }
    i++;
  }
```

```
  System.out.println(total);
}
```

16. Solution

```
public static void main(String[] args) {
  int a, i, total;

  total = 0;

  i = 1;
  while (i <= 20) {
    a = Integer.parseInt(cin.nextLine());
    if (a >= 100 && a <= 999) {
      total = total + a;
    }
    i++;
  }
  System.out.println(total);
}
```

17. Solution

```
public static void main(String[] args) {
  double a, p;

  p = 1;

  a = Double.parseDouble(cin.nextLine());
  while (a != 0) {
    p = p * a;
    a = Double.parseDouble(cin.nextLine());
  }
  System.out.println(p);
}
```

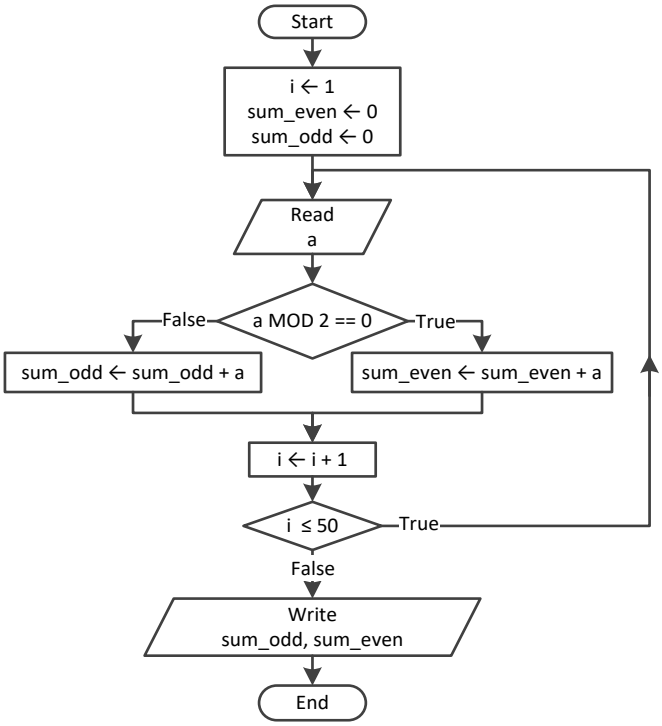
Step	Statement	a	p
1	`p = 1`	?	**1.0**
2	`a = Double.parseDouble(cin.nextLine())`	**3.0**	1.0
3	`while (a != 0)`	true	
4	`p = p * a`	3.0	**3.0**
5	`a = Double.parseDouble(cin.nextLine())`	**2.0**	3.0
6	`while (a != 0)`	true	
7	`p = p * a`	2.0	**6.0**
8	`a = Double.parseDouble(cin.nextLine())`	**9.0**	6.0
9	`while (a != 0)`	true	
10	`p = p * a`	9.0	**54.0**
11	`a = Double.parseDouble(cin.nextLine())`	**0.0**	54.0

12	`while (a != 0)`	false
13	`System.out.println(p)`	It displays: 54.0

18. Solution

```
public static void main(String[] args) {
  int years;
  double population;

  population = 30000;

  years = 0;
  while (population <= 100000) {
    population += population * 0.03;
    years++;
  }
  System.out.println(years);
}
```

19. Solution




```
public static void main(String[] args) {
  int a, i, sum_even, sum_odd;

  i = 1;
  sum_even = 0;
  sum_odd = 0;
  do {
    a = Integer.parseInt(cin.nextLine());
    if (a % 2 == 0) {
```

```
      sum_even += a;
    }
    else {
      sum_odd += a;
    }
    i++;
  } while (i <= 50);
  System.out.println(sum_even + " " + sum_odd);
}
```

20. Solution

```
public static void main(String[] args) {
  int a, i, n, p;

  n = Integer.parseInt(cin.nextLine());
  i = 1;
  p = 1;
  do {
    a = Integer.parseInt(cin.nextLine());
    if (a < 0) {
      p *= a;
    }
    i++;
  } while (i <= n);
  System.out.println(Math.abs(p));
}
```

21. Solution

```
public static void main(String[] args) {
  int a, i, p;

  i = 1;
  p = 1;
  do {
    System.out.print("Enter an integer: ");
    a = Integer.parseInt(cin.nextLine());
    if (a >= 500 && a <= 599) {
      p *= a;
    }
    i++;
  } while (i <= 5);
  System.out.println(p);
}
```

22. Solution

```
public static void main(String[] args) {
  double population;
  int years;
```

```
  population = 50000;

  years = 0;
  do {
    population -= population * 0.10;
    years++;
  } while (population >= 20000);
  System.out.println(years);
}
```

Chapter 26

26.3 Review Questions: True/False

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

26.4 Review Questions: Multiple Choice

1. c
2. d
3. d
4. b
5. a
6. b
7. a
8. b
9. c
10. b
11. d
12. d
13. 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)`	true		
6	`b++`	0	**1**	0
7	`j += 2`	0	1	**2**
8	`j <= 8`	true		
9	`if (j < 5)`	true		
10	`b++`	0	**2**	2
11	`j += 2`	0	2	**4**
12	`j <= 8`	true		
13	`if (j < 5)`	true		
14	`b++`	0	**3**	4
15	`j += 2`	0	3	**6**
16	`j <= 8`	true		
17	`if (j < 5)`	false		
18	`a += j - 1`	**5**	3	6

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

2. *Solution*

For input value of 10

Step	Statement	a	b	j
1	`a = Integer.parseInt(cin.nextLine())`	**10**	?	?
2	`b = a`	10	**10**	?
3	`j = a - 5`	10	10	**5**
4	`j <= a`	true		
5	`if (j % 2 != 0)`	true		
6	`b = a + j + 5`	10	**20**	5
7	`j += 2`	10	20	**7**
8	`j <= a`	true		
9	`if (j % 2 != 0)`	true		
10	`b = a + j + 5`	10	**22**	7
11	`j += 2`	10	22	**9**
12	`j <= a`	true		
13	`if (j % 2 != 0)`	true		
14	`b = a + j + 5`	10	**24**	9
15	`j += 2`	10	24	**11**
16	`j <= a`	false		
17	`System.out.println(b)`	It displays: 24		

For input value of 21

Step	Statement	a	b	j
1	`a = Integer.parseInt(cin.nextLine())`	**21**	?	?
2	`b = a`	21	**21**	?
3	`j = a - 5`	21	21	**16**
4	`j <= a`	true		
5	`if (j % 2 != 0)`	false		
6	`b = a + j + 5`	21	**5**	16
7	`j += 2`	21	5	**18**

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

3. Solution

For input value of 12

Step	Statement	a	x	y	j
1	`a = Integer.parseInt(cin.nextLine())`	**12**	?	?	?
2	`j = 2`	12	?	?	**2**
3	`j <= a - 1`	true			
4	`x = j * 3 + 3`	12	**9**	?	2
5	`y = j * 2 + 10`	12	9	**14**	2
6	`if (y - x > 0 || x > 30)`	true			
7	`y *= 2`	12	9	**28**	2
8	`x += 4`	12	**13**	28	2
9	`System.out.println(x + ", " + y)`	It displays: 13, 28			
10	`j += 3`	12	13	28	**5**
11	`j <= a - 1`	true			
12	`x = j * 3 + 3`	12	**18**	28	5
13	`y = j * 2 + 10`	12	18	**20**	5
14	`if (y - x > 0 || x > 30)`	true			
15	`y *= 2`	12	18	**40**	5
16	`x += 4`	12	**22**	40	5
17	`System.out.println(x + ", " + y)`	It displays: 22, 40			
18	`j += 3`	12	22	40	**8**
19	`j <= a - 1`	true			
20	`x = j * 3 + 3`	12	**27**	40	8
21	`y = j * 2 + 10`	12	27	**26**	8
22	`if (y - x > 0 || x > 30)`	false			
23	`x += 4`	12	**31**	26	8

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

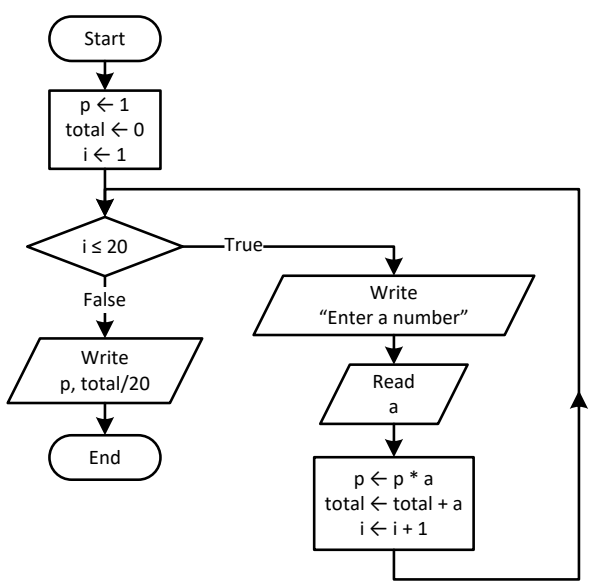
4. Solution

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




```
public static void main(String[] args) {
  double a, p, total;
  int i;

  p = 1;
```

```
  total = 0;
  for (i = 1 ; i <= 20; i++) {
    System.out.print("Enter a number: ");
    a = Double.parseDouble(cin.nextLine());
    p = p * a;
    total = total + a;
  }
  System.out.println(p);
  System.out.println(total / 20);
}
```

7. Solution

```
public static void main(String[] args) {
  double i;

  for (i = 0 ; i <= 360; i += 0.5) {
    System.out.println(Math.sin(i * Math.PI / 180));
  }
}
```

8. Solution

```
public static void main(String[] args) {
  int deg, i;

  System.out.print("Enter degrees: ");
  deg = Integer.parseInt(cin.nextLine());
  for (i = 0 ; i <= deg; i++) {
    System.out.println(Math.cos(i * Math.PI / 180));
  }
}
```

9. Solution

```
public static void main(String[] args) {
  int i, s;

  s = 0;
  for (i = 1; i <= 99; i += 2) {
    s += i;
  }
  System.out.println(s);
}
```

10. Solution

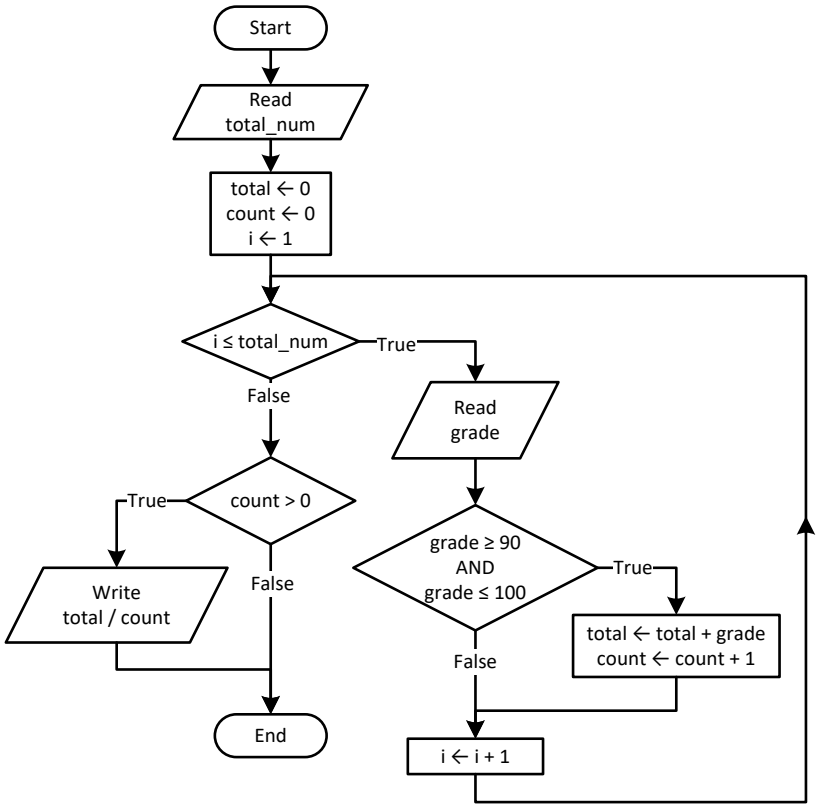
```
public static void main(String[] args) {
  int i, n;
  double p;

  n = Integer.parseInt(cin.nextLine());
```

```
  p = 1;
  for (i = 2; i <= 2 * n; i += 2) {
    p *= Math.pow(i, i - 1);
  }
  System.out.println(p);
}
```

11. Solution

```
public static void main(String[] args) {
  int i, offset, s;

  s = 0;
  i = 1;
  offset = 0;
  while (i <= 191) {
    s += i;
    offset++;
    i += offset;
  }
  System.out.println(s);
}
```

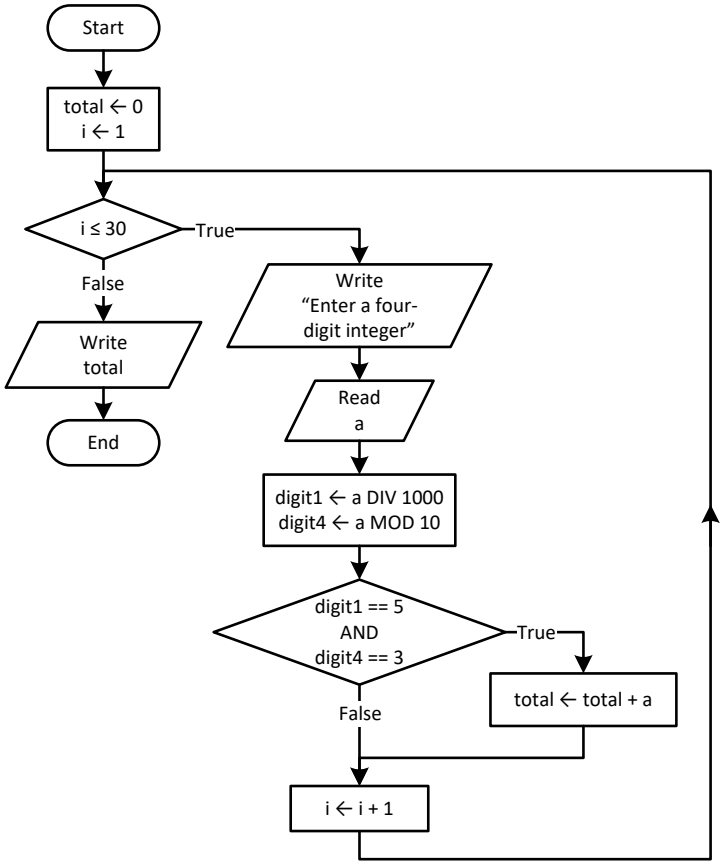
12. Solution




```
public static void main(String[] args) {
  int count, grade, i, total_num, total;
```

```
  total_num = Integer.parseInt(cin.nextLine());
  total = 0;
  count = 0;
  for (i = 1; i <= total_num; i++) {
    grade = Integer.parseInt(cin.nextLine());
    if (grade >= 90 && grade <= 100) {
      total += grade;
      count++;
    }
  }
  if (count > 0) {
    System.out.println(total / (double)count);
  }
}
```

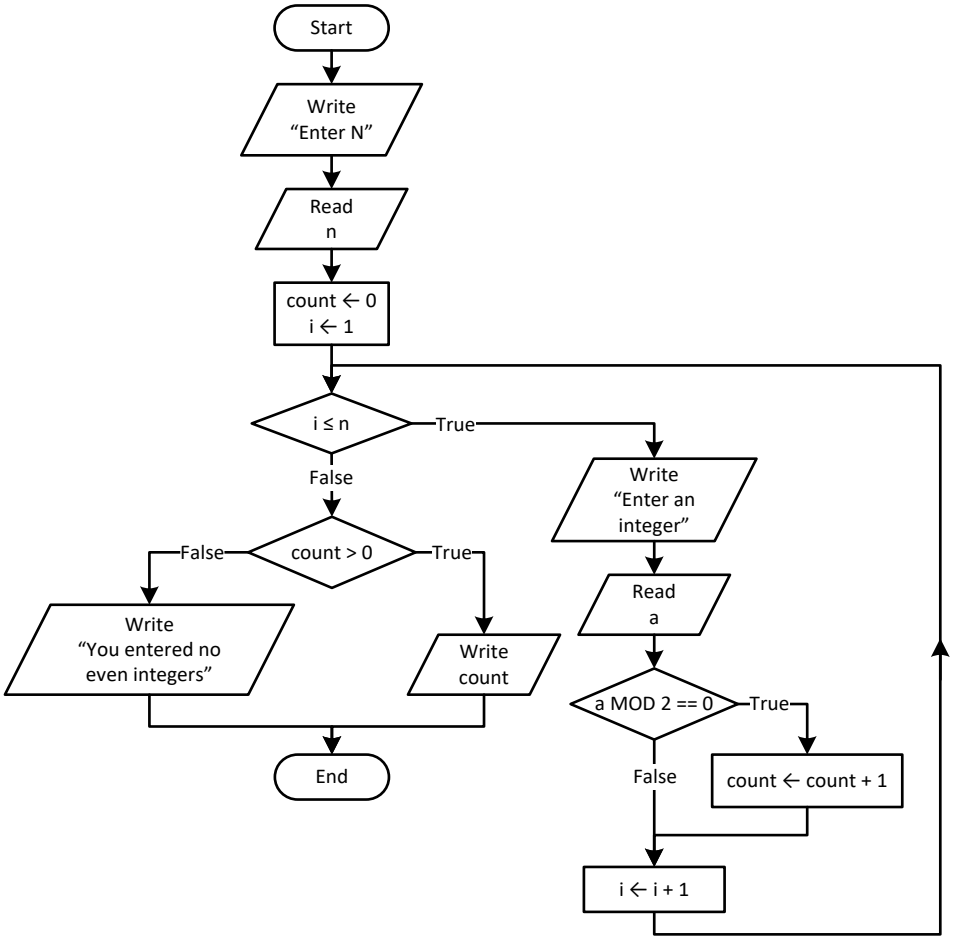
13. Solution




```
public static void main(String[] args) {
  int a, digit1, digit4, i, total;

  total = 0;
  for (i = 1; i <= 30; i++) {
    System.out.print("Enter a four-digit integer: ");
    a = Integer.parseInt(cin.nextLine());
    digit1 = (int)(a / 1000);
    digit4 = a % 10;
```

```
    if (digit1 == 5 && digit4 == 3) {
      total += a;
    }
  }
  System.out.println(total);
}
```

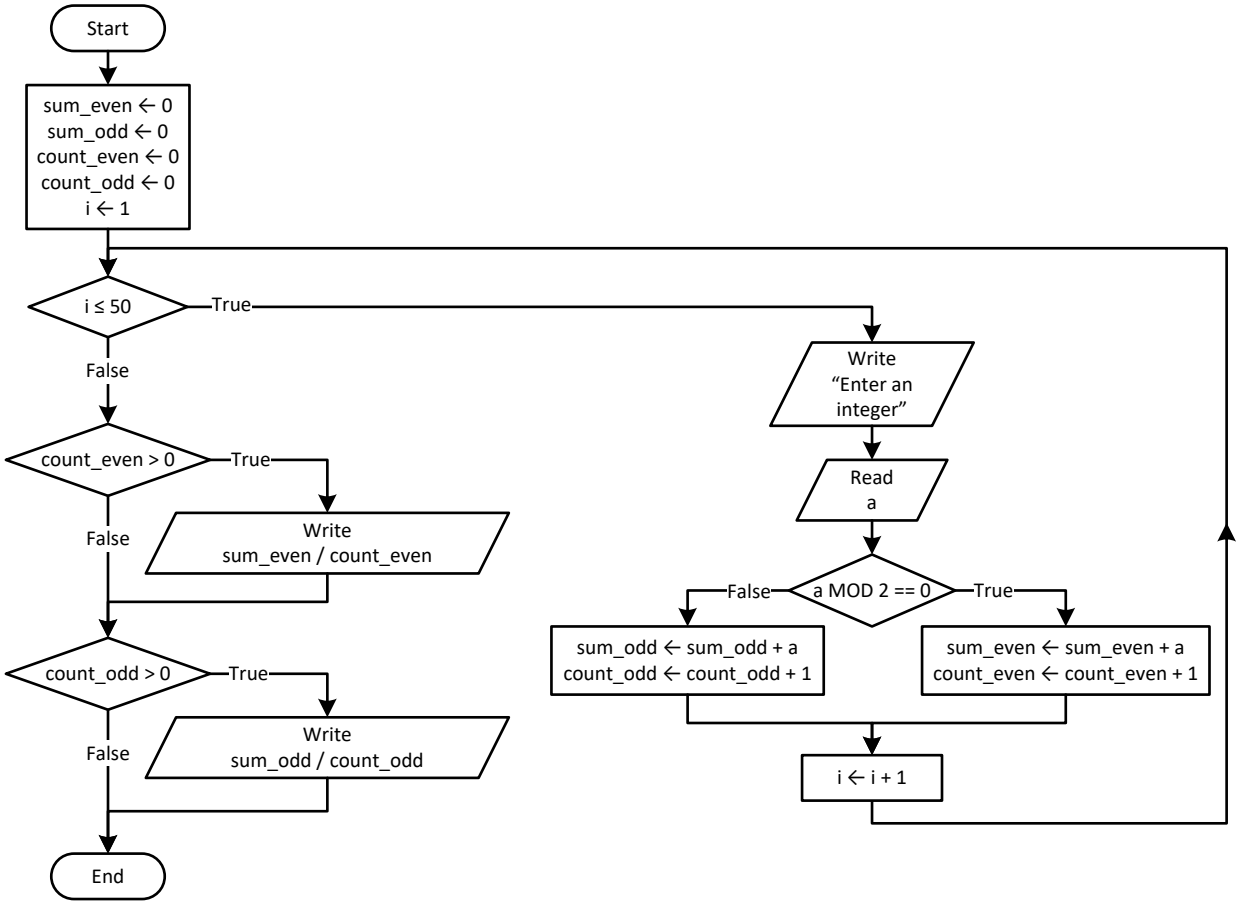
14. Solution




```
public static void main(String[] args) {
  int a, count, i, n;

  System.out.print("Enter N: ");
  n = Integer.parseInt(cin.nextLine());
  count = 0;
  for (i = 1; i <= n; i++) {
    System.out.print("Enter an integer: ");
    a = Integer.parseInt(cin.nextLine());
    if (a % 2 == 0) {
      count++;
    }
  }
  if (count > 0) {
```

```
    System.out.println(count);
  }
  else {
    System.out.println("You entered no even integers");
  }
}
```

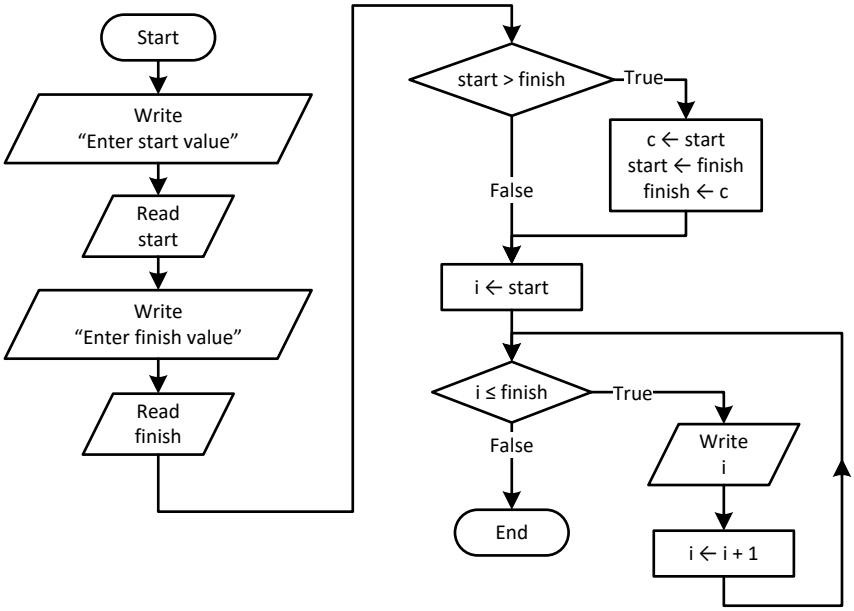
15. Solution




```
public static void main(String[] args) {
  int a, count_even, count_odd, i, sum_even, sum_odd;

  sum_even = 0;
  sum_odd = 0;
  count_even = 0;
  count_odd = 0;
  for (i = 1; i <= 50; i++) {
    System.out.print("Enter an integer: ");
    a = Integer.parseInt(cin.nextLine());
    if (a % 2 == 0) {
      sum_even += a;
      count_even++;
    }
    else {
      sum_odd += a;
```

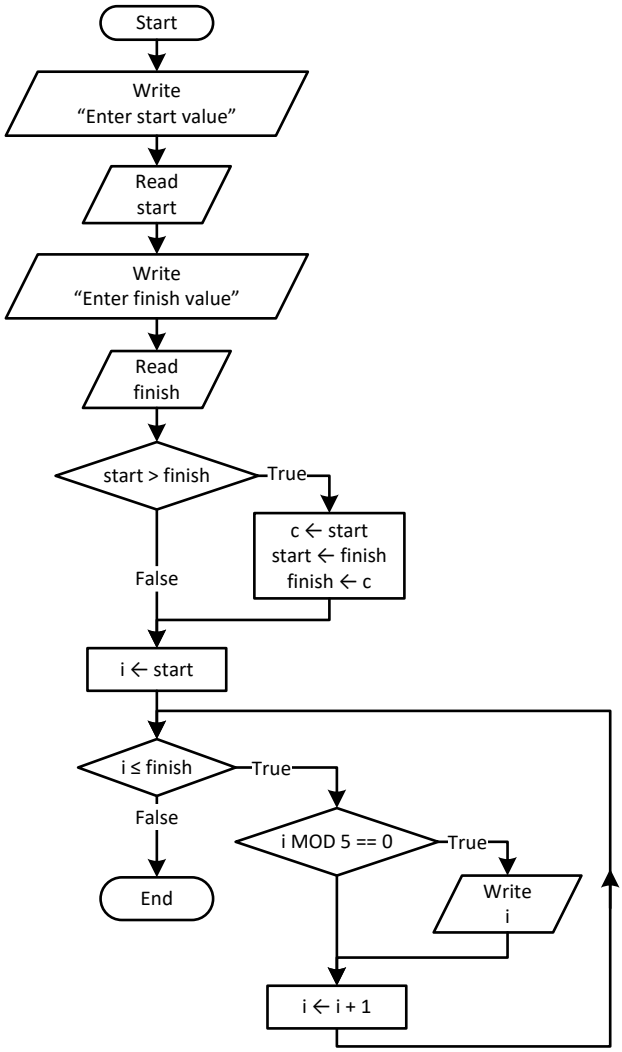
```
      count_odd++;
    }
  }
  if (count_even > 0) {
    System.out.println(sum_even / (double)count_even);
  }
  if (count_odd > 0) {
    System.out.println(sum_odd / (double)count_odd);
  }
}
```

16. Solution




```
public static void main(String[] args) {
  int c, finish, i, start;

  System.out.print("Enter start value: ");
  start = Integer.parseInt(cin.nextLine());
  System.out.print("Enter finish value: ");
  finish = Integer.parseInt(cin.nextLine());

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

  for (i = start; i <= finish; i++) {
    System.out.println(i);
  }
}
```

17. Solution




```
public static void main(String[] args) {
  int c, finish, i, start;

  System.out.print("Enter start value: ");
  start = Integer.parseInt(cin.nextLine());
  System.out.print("Enter finish value: ");
  finish = Integer.parseInt(cin.nextLine());

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

  for (i = start; i <= finish; i++) {
    if (i % 5 == 0) {
      System.out.println(i);
    }
```

```
  }
}
```

18. Solution

First approach

```
public static void main(String[] args) {
  int exp, i;
  double p, b;

  System.out.print("Enter a value for base: ");
  b = Double.parseDouble(cin.nextLine());
  System.out.print("Enter an integer for exponent: ");
  exp = Integer.parseInt(cin.nextLine());

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

Second approach

```
public static void main(String[] args) {
  int exp, i;
  double p, b;

  System.out.print("Enter a value for base: ");
  b = Double.parseDouble(cin.nextLine());
  System.out.print("Enter an integer for exponent: ");
  exp = Integer.parseInt(cin.nextLine());

  p = 1;
  for (i = 1; i <= Math.abs(exp); i++) {
    p *= b;
  }
  if (exp < 0) {
    p = 1 / p;
  }
  System.out.println(p);
}
```

19. Solution

```
public static void main(String[] args) {
```

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

  System.out.print("Enter a message: ");
  msg = cin.nextLine();

  count = 0;
  for (i = 0; i <= msg.length() - 1; i++) {
    character = "" + msg.charAt(i);
    if (character.equals(" ")) {
      count++;
    }
  }
  words = count + 1;

  System.out.println("The message entered contains " + words + " words");
}
```

20. Solution

```
public static void main(String[] args) {
  int characters, count, i, words;
  String msg, character;

  System.out.print("Enter a message: ");
  msg = cin.nextLine();

  characters = msg.length();
  count = 0;
  for (i = 0; i <= characters - 1; i++) {
    character = "" + msg.charAt(i);
    if (character.equals(" ")) {
      count++;
    }
  }

  words = count + 1;
  System.out.print("The average number of letters in each word is ");
  System.out.println((characters - count) / (double)words);
}
```

21. Solution

```
public static void main(String[] args) {
  String message;
  char character;
  String consonants = "BCDFGHJKLMNPQRSTVWZYX";
  int i, count;

  System.out.print("Enter an English message: ");
  message = cin.nextLine().toUpperCase();
```

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

    if (consonants.indexOf(character) != -1) { //If character is found in consonants
      count++;
    }
  }
  System.out.println("Consonants: " + count);
}
```

22. Solution

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

  System.out.print("Enter an English message: ");
  message = cin.nextLine().toUpperCase();

  countv = countc = countd = 0;
  for (i = 0; i <= message.length() - 1; i++) {
    character = message.charAt(i);

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

Chapter 27

27.3 Review Questions: True/False

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

27.4 Review Questions: Multiple Choice

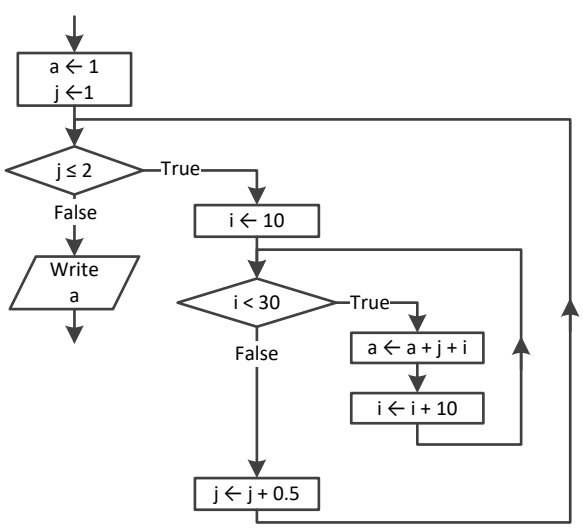
1. b
2. a
3. c
4. a
5. b

27.5 Review Exercises

1. Solution

i. 10
ii. A value greater than or equal to 4.5 and less than 5.0
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	`System.out.println(a)`	It displays: 100		

3. Solution

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

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

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

4. Solution

For input value of “NO”

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

For input values of "YES", "NO"

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

16	`i <= 3`	false			
17	`ans = cin.nextLine()`	61	10	4	**"YES"**
18	`while (ans.equals("YES"))`	true			
19	`i = 1`	61	10	**1**	"YES"
20	`i <= 3`	true			
21	`s = s + y`	**71**	10	1	"YES"
22	`y -= 5`	71	**5**	1	"YES"
23	`i++`	71	5	**2**	"YES"
24	`i <= 3`	true			
25	`s = s + y`	**76**	5	2	"YES"
26	`y -= 5`	76	**0**	2	"YES"
27	`i++`	76	0	**3**	"YES"
28	`i <= 3`	true			
29	`s = s + y`	**76**	0	3	"YES"
30	`y -= 5`	76	**–5**	3	"YES"
31	`i++`	76	–5	**4**	"YES"
32	`i <= 3`	false			
33	`ans = cin.nextLine()`	76	–5	4	**"NO"**
34	`while (ans.equals("YES"))`	false			
35	`System.out.println(s)`	It displays: 76			

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

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

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

5. Solution

```
public static void main(String[] args) {
  int hour, minutes;

  for (hour = 0; hour <= 23; hour++) {
    for (minutes = 0; minutes <= 59; minutes++) {
      System.out.println(hour + "\t" + minutes);
    }
  }
}
```

6. Solution

```
public static void main(String[] args) {
  int i, j;

  for (i = 5; i >= 1; i--) {
    for (j = 1; j <= i; j++) {
      System.out.print(i + " ");
    }
    System.out.println();
  }
}
```

7. Solution

```
public static void main(String[] args) {
  int i, j;

  for (i = 0; i <= 5; i++) {
    for (j = 0; j <= i; j++) {
      System.out.print(j + " ");
    }
    System.out.println();
  }
}
```

8. Solution

```
public static void main(String[] args) {
  int i, j;

  for (i = 1; i <= 4; i++) {
    for (j = 1; j <= 10; j++) {
      System.out.print("* ");
    }
    System.out.println();
  }
}
```

9. Solution

```
public static void main(String[] args) {
  int i, j, y;

  System.out.print("Enter an integer between 3 and 20: ");
  y = Integer.parseInt(cin.nextLine());

  for (i = 1; i <= y; i++) {
    for (j = 1; j <= y; j++) {
      System.out.print("* ");
    }
    System.out.println();
  }
}
```

10. Solution

```
public static void main(String[] args) {
  int i, j, y;

  System.out.print("Enter an integer between 3 and 20: ");
  y = Integer.parseInt(cin.nextLine());

  for (j = 1; j <= y; j++) {
    System.out.print("* ");
  }
  System.out.println();

  for (i = 1; i <= y - 2; i++) {
    System.out.print("* ");
    for (j = 1; j <= y - 2; j++) {
      System.out.print("  ");
    }
    System.out.println("* ");
  }

  for (j = 1; j <= y; j++) {
    System.out.print("* ");
  }
}
```

11. Solution

```
public static void main(String[] args) {
  int i, j;

  for (i = 1; i <= 5; i++) {
    for (j = 1; j <= i; j++) {
      System.out.print("* ");
    }
```

```
    System.out.println();
  }

  for (i = 4; i >= 1; i--) {
    for (j = 1; j <= i; j++) {
      System.out.print("* ");
    }
    System.out.println();
  }
}
```

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 = "";
System.out.print("Enter a name: ");
name = cin.nextLine();
while (!name.equals("STOP")) {
  System.out.print("Enter a name: ");
  name = cin.nextLine();
  count_names++;
  if (!name.equals("John")) {
    count_not_johns++;
  }
  System.out.print("Enter a name: ");
  name = cin.nextLine();
}
System.out.println(count_names + " names entered");
System.out.println("Names other than John entered " + count_not_johns + " times");
```

2. Solution

First approach

```
public static void main(String[] args) {
  String text, character;
  boolean found;
  int i;

  System.out.print("Enter a text: ");
  text = cin.nextLine();

  found = false;
```

```
  for (i = 0; i <= text.length() - 1; i++) {
    character = "" + text.charAt(i);
    if (character.equals(" ")) {
      found = true;
      break;
    }
  }

  if (!found) {
    System.out.println("One Single Word");
  }
  else {
    System.out.println("Complete Sentence");
  }
}
```

Second approach

```
public static void main(String[] args) {
  String text;

  System.out.print("Enter a text: ");
  text = cin.nextLine();

  if (text.indexOf(" ") == -1) {
    System.out.println("One Single Word");
  }
  else {
    System.out.println("Complete Sentence");
  }
}
```

3. Solution

First approach

```
public static void main(String[] args) {
  String sentence, character;
  boolean found;
  int i;
  String digits = "0123456789";

  System.out.print("Enter a text: ");
  sentence = cin.nextLine();

  found = false;
  for (i = 0; i <= sentence.length() - 1; i++) {
    character = "" + sentence.charAt(i);
    if (digits.indexOf(character) != -1) {
      found = true;
      break;
    }
  }
```

```
  if (found) {
    System.out.println("The sentence contains a number");
  }
}
```

Second approach

```
public static void main(String[] args) {
  String sentence;
  boolean found;
  int i;
  String digit;

  System.out.print("Enter a text: ");
  sentence = cin.nextLine();

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

  if (found) {
    System.out.println("The sentence contains a number");
  }
}
```

4. Solution

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

5. Solution

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

6. Solution

```
s = 0;
for (i = 1; i <= 100; i++) {
  number = Double.parseDouble(cin.nextLine());
  s = s + number;
}
```

```
average = s / 100.0;
System.out.println(average);
```

7. Solution

```
int i, denom;
double s;

s = 0;

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

for (i = 1; i <= 100; i++) {
  s += i / (double)denom;
}
System.out.println(s);
```

8. Solution

```
public static void main(String[] args) {
  int i, j;

  for (i = 1; i <= 4; i++) {
    for (j = 1; j <= 4; j++) {
      System.out.println(i + " x " + j + " = " + (i * j));
    }
  }
}
```

9. Solution

```
public static void main(String[] args) {
  int i, j;

  System.out.print("\t|\t");
  for (i = 1; i <= 12; i++) {
    System.out.print(i + "\t");
  }
  System.out.println();

  for (i = 1; i <= 12; i++) {
    System.out.print("---------");
  }
  System.out.println();

  for (i = 1; i <= 12; i++) {
    System.out.print(i + "\t|\t");
    for (j = 1; j <= 12; j++) {
      System.out.print(i * j + "\t");
```

```
    }
    System.out.println();
  }
}
```

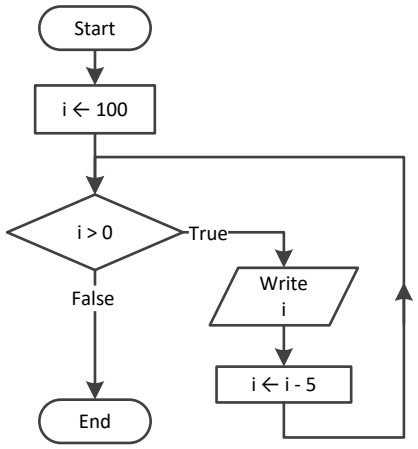
10. Solution

```
public static void main(String[] args) {
  int i, j, n;

  System.out.print("Enter an integer: ");
  n = Integer.parseInt(cin.nextLine());

  System.out.print("\t|\t");
  for (i = 1; i <= n; i++) {
    System.out.print(i + "\t");
  }
  System.out.println();

  for (i = 1; i <= n; i++) {
    System.out.print("---------");
  }
  System.out.println();

  for (i = 1; i <= n; i++) {
    System.out.print(i + "\t|\t");
    for (j = 1; j <= n; j++) {
      System.out.print(i * j + "\t");
    }
    System.out.println();
  }
}
```

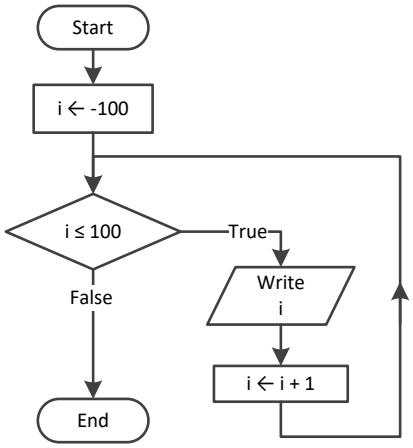
Chapter 29

29.4 Review Exercises

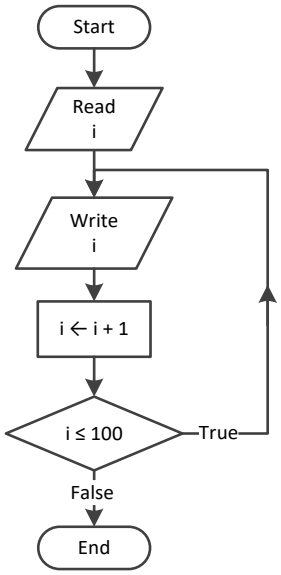
1. Solution



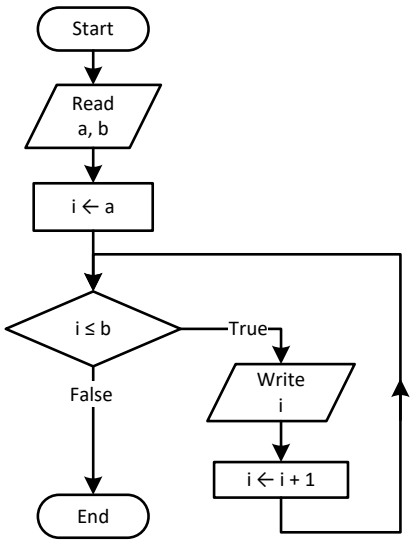
2. Solution



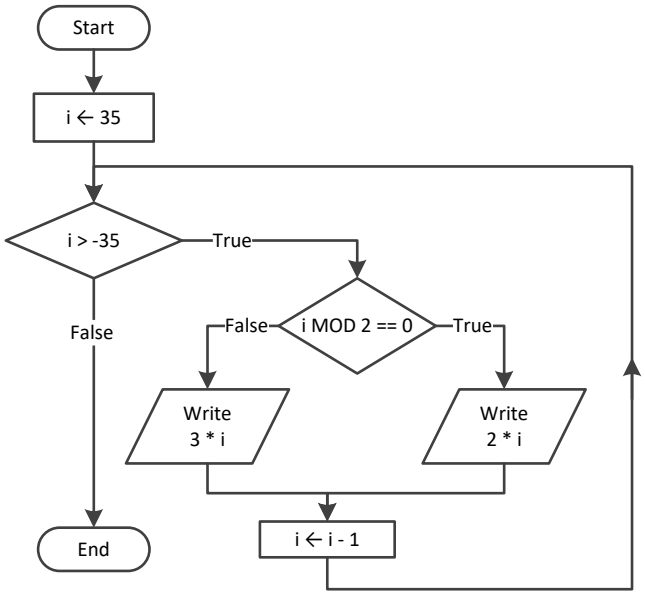
3. Solution



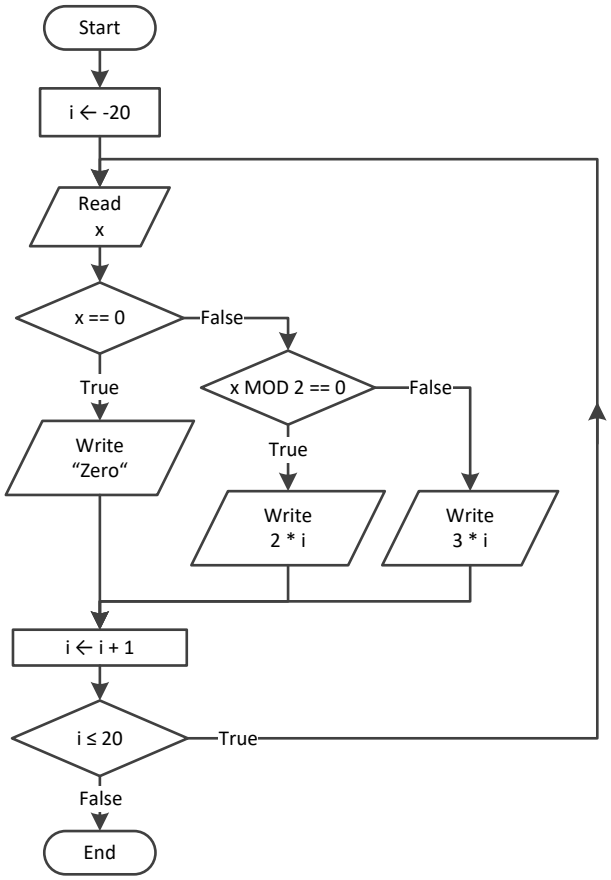
4. Solution



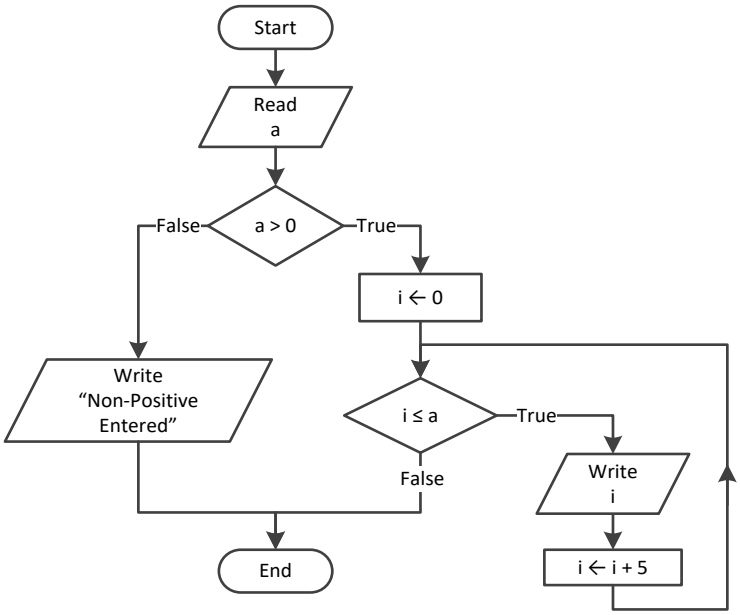
5. Solution



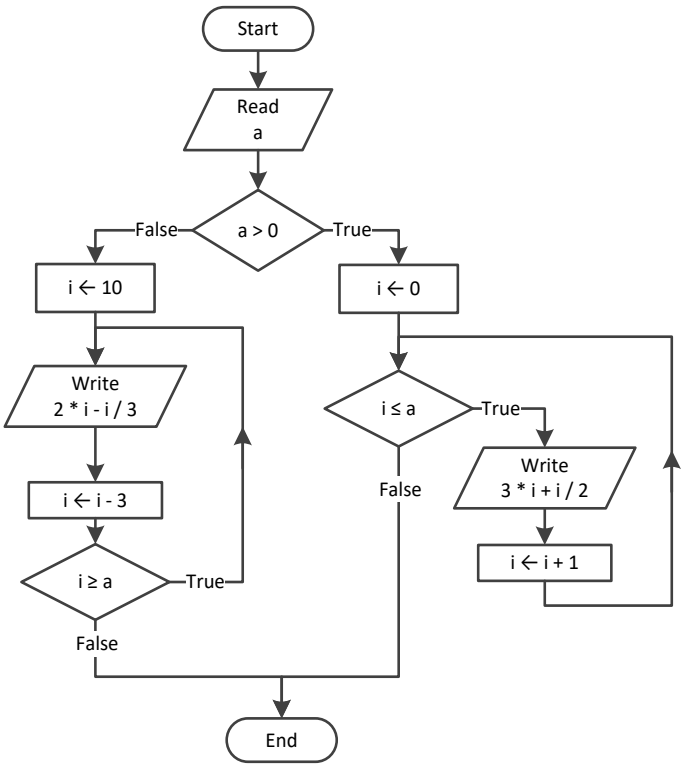
6. Solution



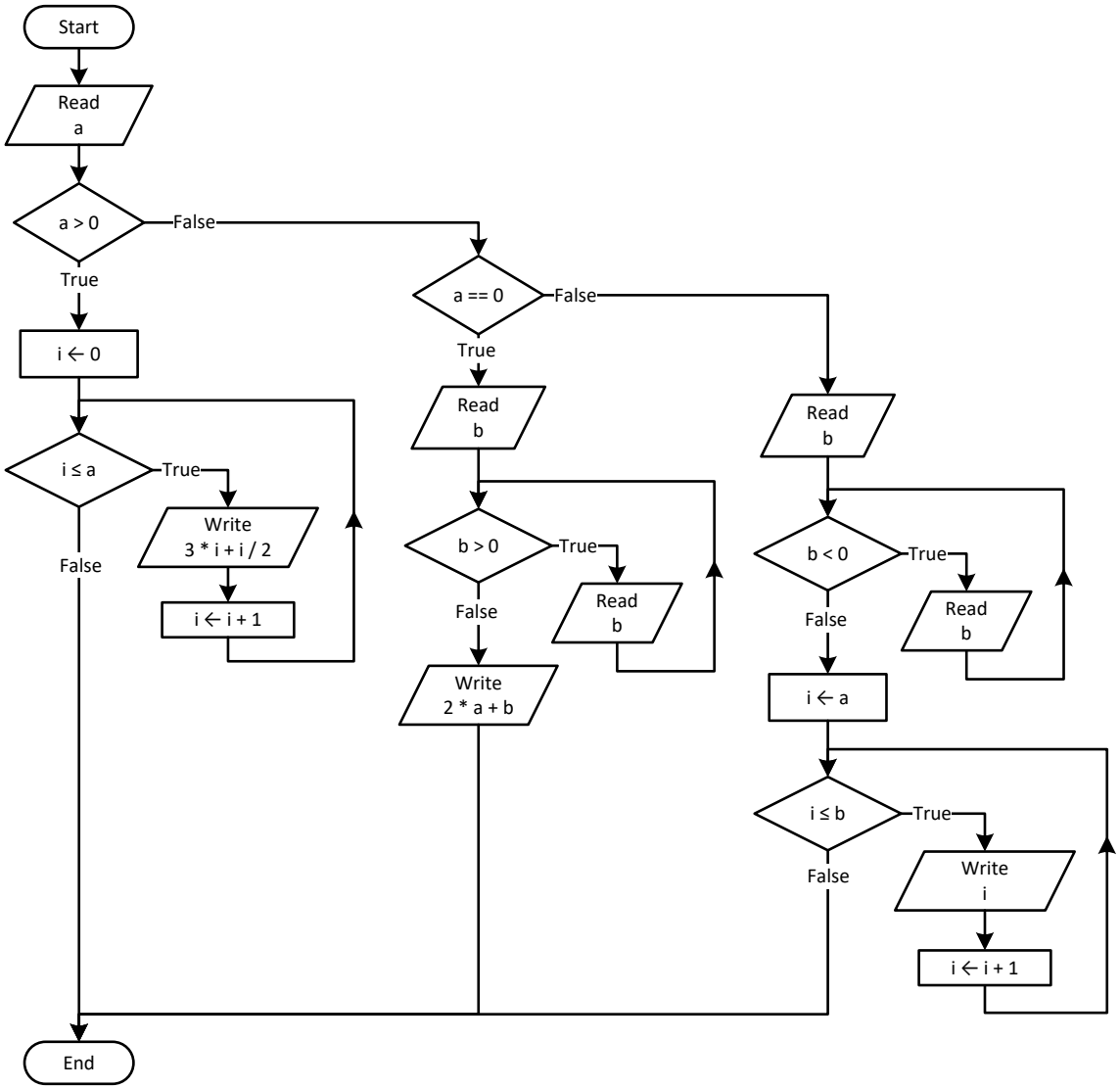
7. Solution



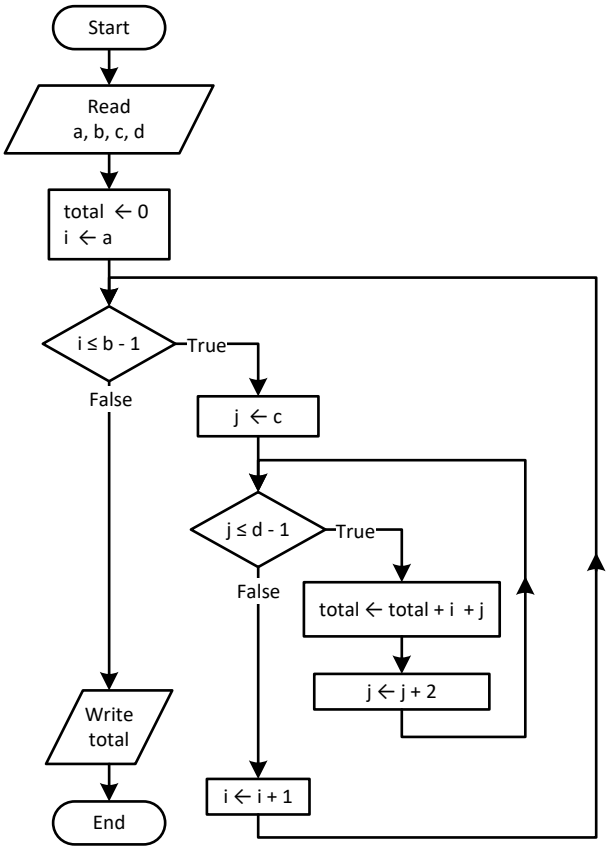
8. Solution



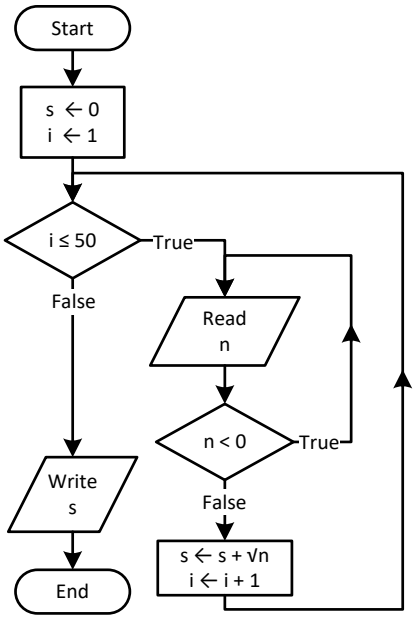
9. Solution


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

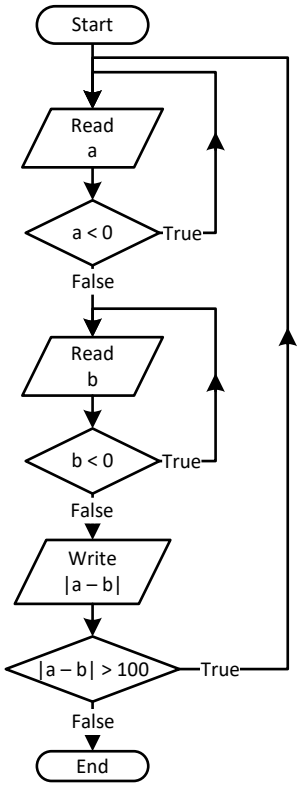
10. Solution



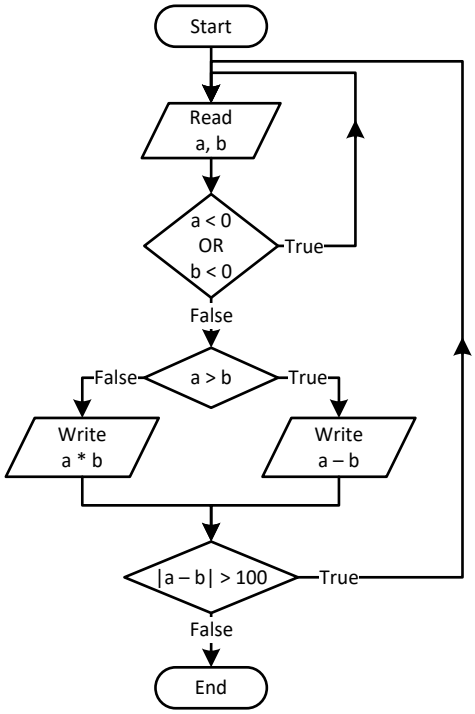
11. Solution



12. Solution



13. Solution



14. Solution

```
i = 1;
do {
  System.out.println(i);
  i += 5;
} while (i <= 500);
System.out.println("The End");
```

15. Solution

```
public static void main(String[] args) {
  int a, i;

  i = 0;
  a = Integer.parseInt(cin.nextLine());
  do {
    if (i % 2 != 0) {
      System.out.println(i);
    }
    i += 5;
  } while (i < a);
}
```

16. Solution

```
public static void main(String[] args) {
  int a, b, i;

  a = Integer.parseInt(cin.nextLine());
  while (a != -1) {
    do {
      b = Integer.parseInt(cin.nextLine());
    } while (b <= a);
    for (i = a; i <= b; i++) {
      System.out.println(i);
    }
    a = Integer.parseInt(cin.nextLine());
  }
}
```

17. Solution

```
public static void main(String[] args) {
  int i;
  double P, S, a;

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

```
  a = 0;

  while (true) {
    if (i < 45) {
      S += a;
    }
    else {
      P *= a;
    }
    i++;
    if (i >= 90) break;
    a = Double.parseDouble(cin.nextLine());
  }

  System.out.println(S + " " + P);
}
```

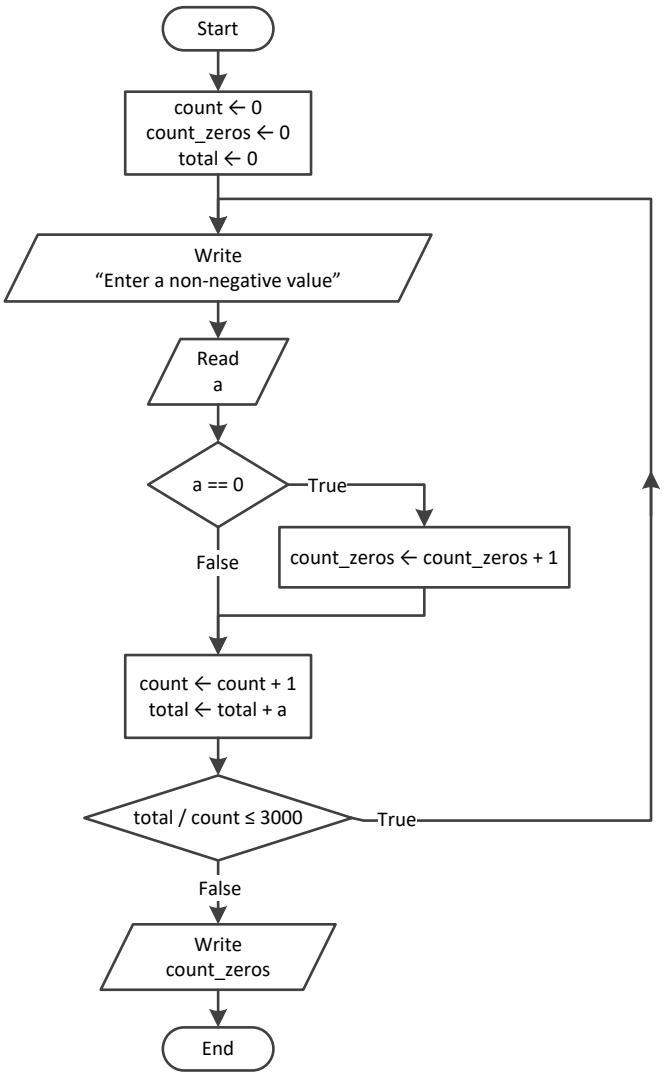
Chapter 30

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




```
public static void main(String[] args) {
  int count, count_zeros;
  double a, total;

  count = 0;
  count_zeros = 0;
```

```
  total = 0;
  do {
    System.out.print("Enter a non-negative value: ");
    a = Double.parseDouble(cin.nextLine());
    if (a == 0) {
      count_zeros++;
    }
    count++;
    total += a;
  } while (total / count <= 3000);
  System.out.println(count_zeros);
}
```

2. Solution

First approach

```
public static void main(String[] args) {
  int a, d1, d2, d3, d4, i, r;

  System.out.print("Enter an integer between 1 and 20: ");
  a = Integer.parseInt(cin.nextLine());
  for (i = 1000; i <= 9999; i++) {
    d4 = i % 10;
    r = (int)(i / 10);
    d3 = r % 10;
    r = (int)(r / 10);
    d2 = r % 10;
    d1 = (int)(r / 10);
    if (d1 + d2 + d3 + d4 < a) {
      System.out.println(i);
    }
  }
}
```

Second approach

```
public static void main(String[] args) {
  int a, d1, d2, d3, d4;

  System.out.print("Enter an integer between 1 and 20: ");
  a = Integer.parseInt(cin.nextLine());
  for (d1 = 1; d1 <= 9; d1++) {
    for (d2 = 0; d2 <= 9; d2++) {
      for (d3 = 0; d3 <= 9; d3++) {
        for (d4 = 0; d4 <= 9; d4++) {
          if (d1 + d2 + d3 + d4 < a) {
            System.out.println(d1 * 1000 + d2 * 100 + d3 * 10 + d4);
          }
        }
      }
    }
  }
}
```

3. Solution

First approach

```
public static void main(String[] args) {
  int d1, d2, d3, d4, i, r;

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

Second approach

```
public static void main(String[] args) {
  int d1, d2, d3, d4;

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

4. Solution

First approach

```
public static void main(String[] args) {
  int x, count;

  System.out.print("Enter a number: ");
  x = Integer.parseInt(cin.nextLine());

  count = 0;

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

```
  System.out.println(count);
}
```

Second approach

```
public static void main(String[] args) {
  int x, count;

  System.out.print("Enter a number: ");
  x = Integer.parseInt(cin.nextLine());

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

  System.out.println(count);
}
```

5. Solution

```
x = Integer.parseInt(cin.nextLine());
while (x != 1 && x != 0) {
  System.out.println("Error");
  x = Integer.parseInt(cin.nextLine());
}
```

6. Solution

```
do {
  gender = cin.nextLine().toUpperCase();
} while (!gender.equals("M") && !gender.equals("F"));
```

7. Solution

```
public static void main(String[] args) {
  int count, x;
  double y;

  System.out.print("Enter a non-negative number: ");
  x = Integer.parseInt(cin.nextLine());
  count = 0;
  while (x < 0) {
    count++;
    if (count == 2) break;

    System.out.println("Error: Invalid number!");
    System.out.print("Enter a non-negative number: ");
    x = Integer.parseInt(cin.nextLine());
  }

  if (count < 2) {
    y = Math.sqrt(x);
    System.out.println(y);
```

```
  }
  else {
    System.out.println("Dude, you are dumb!");
  }
}
```

8. Solution

```
public static void main(String[] args) {
  String answer;
  double area, r;

  do {
    System.out.print("Enter the length of a radius: ");
    r = Double.parseDouble(cin.nextLine());
    while (r <= 0) {
      System.out.print("Invalid radius. Enter the length of a radius: ");
      r = Double.parseDouble(cin.nextLine());
    }

    area = Math.PI * Math.pow(r, 2);
    System.out.println("The area is: " + area);

    System.out.print("Would you like to repeat? ");
    answer = cin.nextLine();
  } while (answer.toUpperCase().equals("YES"));

}
```

9. Solution

```
public static void main(String[] args) {
  int x, y;

  for (x = -100; x <= 100; x++) {
    for (y = -100; y <= 100; y++) {
      if (5 * x + 3 * Math.pow(y, 2) == 0) {
        System.out.println(x + ", " + y);
      }
    }
  }
}
```

10. Solution

```
public static void main(String[] args) {
  int x, y, z;

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

```
            System.out.println(x + ", " + y + ", " + z);
          }
        }
      }
    }
  }
```

11. Solution

```
public static void main(String[] args) {
  int m1, m2, m3, s;

  m1 = Integer.parseInt(cin.nextLine());
  m2 = Integer.parseInt(cin.nextLine());
  m3 = Integer.parseInt(cin.nextLine());

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

  m1 = s;
  m2 = m3;

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

  System.out.println(s);
}
```

12. Solution

```
public static void main(String[] args) {
  int x, number_of_divisors, i;

  x = Integer.parseInt(cin.nextLine());
  while (x <= 0) {
    System.out.println("Error! You must enter a positive integer");
    x = Integer.parseInt(cin.nextLine());
  }
```

```
  number_of_divisors = 2;
  for (i = 2; i <= (int)(x / 2); i++) {
    if (x % i == 0) {
      number_of_divisors++;
    }
  }
  System.out.println(number_of_divisors);
}
```

13. Solution

```
public static void main(String[] args) {
  int x, number_of_divisors, i;

  System.out.print("Enter an integer greater than 1: ");
  x = Integer.parseInt(cin.nextLine());
  while (x <= 1) {
    System.out.println("Error!");
    x = Integer.parseInt(cin.nextLine());
  }

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

  if (number_of_divisors == 2) {
    System.out.println("Number " + x + " is prime");
  }
}
```

14. Solution

```
public static void main(String[] args) {
  int a, b, c, i, number_of_divisors, x;

  System.out.print("Enter an integer greater than 1: ");
  a = Integer.parseInt(cin.nextLine());
  while (a < 2) {
    System.out.print("Wrong number. Please enter an integer greater than 1: ");
    a = Integer.parseInt(cin.nextLine());
  }

  System.out.print("Enter a second integer greater than 1: ");
  b = Integer.parseInt(cin.nextLine());
  while (b < 2) {
    System.out.print("Wrong number. Please enter a second integer greater than 1: ");
    b = Integer.parseInt(cin.nextLine());
```

```
  }

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

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

15. Solution

```
public static void main(String[] args) {
  int a, b, c, d1, d2, d3, d4, r, x;

  System.out.print("Enter a positive four-digit integer: ");
  a = Integer.parseInt(cin.nextLine());
  while (a < 1000 || a > 9999) {
    System.out.print("Wrong number. Please enter a positive four-digit integer: ");
    a = Integer.parseInt(cin.nextLine());
  }

  System.out.print("Enter a second positive four-digit integer: ");
  b = Integer.parseInt(cin.nextLine());
  while (b < 1000 || b > 9999) {
    System.out.print("Wrong number. Please enter a second positive four-digit integer: ");
    b = Integer.parseInt(cin.nextLine());
  }

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

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

```
    d3 = r % 10;
    r = (int)(r / 10);
    d2 = r % 10;
    d1 = (int)(r / 10);

    if (d1 == d4 && d2 == d3) {
      System.out.println(x);
    }
  }
}
```

16. Solution

```
public static void main(String[] args) {
  int i;

  for (i = 0; i <= 30; i++) {
    System.out.println(Math.pow(2, i));
  }
}
```

17. Solution

```
public static void main(String[] args) {
  int i, offset;

  offset = 10;
  i = 1;
  while (i <= 401) {
    System.out.println(i);
    i += offset;
    offset += 2;
  }
}
```

18. Solution

```
public static void main(String[] args) {
  int i;

  for (i = 1; i <= 100; i++) {
    System.out.println(-i + "\n" + i);
  }
}
```

19. Solution

First approach

```
public static void main(String[] args) {
  int i, offset, value;

  value = 0;
```

```
  for (i = 1; i <= 8; i++) {
    offset = (int)Math.pow(10, i - 1);
    value += offset;
    System.out.println(value);
  }
}
```

Second approach

```
public static void main(String[] args) {
  int i;
  String value;

  value = "1";
  for (i = 1; i <= 8; i++) {
    System.out.println(value);
    value += "1";
  }
}
```

20. Solution

```
public static void main(String[] args) {
  int a, fib, fib_prev, fib_prev_prev, i;

  a = Integer.parseInt(cin.nextLine());

  fib_prev_prev = 0;
  fib_prev = 1;
  fib = 1;
  for (i = 1; i <= a; i++) {
    System.out.println(fib);
    fib = fib_prev + fib_prev_prev;
    fib_prev_prev = fib_prev;
    fib_prev = fib;
  }
}
```

21. Solution

```
public static void main(String[] args) {
  int a, fib, fib_prev, fib_prev_prev;

  a = Integer.parseInt(cin.nextLine());

  fib_prev_prev = 0;
  fib_prev = 1;
  fib = 1;
  while (fib < a) {
    System.out.println(fib);
    fib = fib_prev + fib_prev_prev;
    fib_prev_prev = fib_prev;
    fib_prev = fib;
```

```
  }
}
```

22. Solution

```
public static void main(String[] args) {
  int denominator, i, n, nominator;
  double y;

  System.out.print("Enter a positive integer: ");
  n = Integer.parseInt(cin.nextLine());
  while (n <= 0) {
    System.out.print("Wrong number. Please enter a positive integer: ");
    n = Integer.parseInt(cin.nextLine());
  }

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

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

  y = nominator / (double)denominator;
  System.out.println(y);
}
```

23. Solution

```
public static void main(String[] args) {
  int i, n, nominator, sign;
  double y;

  System.out.print("Enter a positive integer: ");
  n = Integer.parseInt(cin.nextLine());
  while (n <= 0) {
    System.out.print("Wrong number. Please enter a positive integer: ");
    n = Integer.parseInt(cin.nextLine());
  }

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

  y = nominator / (double)n;
```

```
  System.out.println(y);
}
```

24. Solution

```
public static void main(String[] args) {
  int i, n, sign;
  double y;

  System.out.print("Enter a positive integer: ");
  n = Integer.parseInt(cin.nextLine());
  while (n <= 0) {
    System.out.print("Wrong number. Please enter a positive integer: ");
    n = Integer.parseInt(cin.nextLine());
  }

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

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

  System.out.println(y);
}
```

25. Solution

```
public static void main(String[] args) {
  int i, n;
  double y;

  System.out.print("Enter a positive integer: ");
  n = Integer.parseInt(cin.nextLine());
  while (n <= 0) {
    System.out.print("Wrong number. Please enter a positive integer: ");
    n = Integer.parseInt(cin.nextLine());
  }

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

  System.out.println(y);
}
```

26. Solution

```
public static void main(String[] args) {
  int factorial, i, n;
```

```
  System.out.print("Enter a non-negative integer: ");
  n = Integer.parseInt(cin.nextLine());

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

  System.out.println(factorial);
}
```

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

27. Solution

First approach

```
static final double ACCURACY = 0.00001;

public static void main(String[] args) {
  int i, j;
  double factorial, exponential_previous, exponential, x;

  x = Double.parseDouble(cin.nextLine());

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

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

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

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

  System.out.println("e(" + x  +") ~= " + exponential);
}
```

Second approach

```
static final double ACCURACY = 0.00001;

public static void main(String[] args) {
  int i;
  double factorial, exponential_previous, exponential, x;

  x = Double.parseDouble(cin.nextLine());
```

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

    factorial *= i;

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

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

  System.out.println("e(" + x + ") ~= " + exponential);
}
```

28. Solution

First approach

```
static final double ACCURACY = 0.00001;

public static void main(String[] args) {
  int i, j, sign;
  double factorial;
  double sinus, sinus_previous, x;

  x = Double.parseDouble(cin.nextLine());

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

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

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

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

  System.out.println("sin(" + x + ") ~= " + sinus);
}
```

Second approach

```
static final double ACCURACY = 0.00001;

public static void main(String[] args) {
```

```
  int i, sign;
  double factorial;
  double sinus, sinus_previous, x;

  x = Double.parseDouble(cin.nextLine());

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

    factorial *= i * (i - 1);

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

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

  System.out.println("sin(" + x + ") ~= " + sinus);
}
```

29. Solution

First approach

```
static final double ACCURACY = 0.00001;

public static void main(String[] args) {
  int i, j, sign;
  double factorial;
  double cosinus, cosinus_previous, x;

  x = Double.parseDouble(cin.nextLine());

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

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

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

    sign = -sign;
    i += 2;
```

```
  } while (Math.abs(cosinus - cosinus_previous) > ACCURACY);

  System.out.println("cos(" + x + ") ~= " + cosinus);
}
```

Second approach

```
static final double ACCURACY = 0.00001;

public static void main(String[] args) {
  int i, sign;
  double factorial;
  double cosinus, cosinus_previous, x;

  x = Double.parseDouble(cin.nextLine());

  sign = -1;
  cosinus = 1;
  i = 2;
  factorial = 1;
  do {
    cosinus_previous = cosinus;

    factorial *= i * (i - 1);

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

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

  System.out.println("cos(" + x + ") ~= " + cosinus);
}
```

30. Solution

```
public static void main(String[] args) {
  int i;
  double maximum, total, t;

  maximum = -460;
  total = 0;
  for (i = 1; i <= 31; i++) {
    System.out.print("Enter temperature for day " + i + ": ");
    t = Double.parseDouble(cin.nextLine());
    while (t < -459.67) {
      System.out.println("Error! Wrong temperature.");
      System.out.print("Enter temperature for day " + i + ": ");
      t = Double.parseDouble(cin.nextLine());
    }

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

```
      maximum = t;
    }
  }

  System.out.println(total / 31 + " " + maximum);
}
```

31. Solution

```
public static void main(String[] args) {
  int hour, max_hour, max_minutes, min_hour, min_minutes, minutes;
  double level, maximum, minimum;

  level = Double.parseDouble(cin.nextLine());
  if (level != 9999) {
    hour = Integer.parseInt(cin.nextLine());
    minutes = Integer.parseInt(cin.nextLine());

    maximum = level;
    max_hour = hour;
    max_minutes = minutes;

    minimum = level;
    min_hour = hour;
    min_minutes = minutes;

    level = Double.parseDouble(cin.nextLine());
    while (level != 9999) {
      hour = Integer.parseInt(cin.nextLine());
      minutes = Integer.parseInt(cin.nextLine());

      if (level > maximum) {
        maximum = level;
        max_hour = hour;
        max_minutes = minutes;
      }

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

      level = Double.parseDouble(cin.nextLine());
    }

    System.out.println(maximum + ", " + max_hour + ", " + max_minutes);
    System.out.println(minimum + ", " + min_hour + ", " + min_minutes);
  }
}
```

32. Solution

```
public static void main(String[] args) {
  int a, b, c, i;
  boolean failure;

  String alphabet = "abcdefghijklmnopqrstuvwxyz";

  do {
    System.out.print("Enter an integer between 1 and 26: ");
    a = Integer.parseInt(cin.nextLine());

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

  do {
    System.out.print("Enter an integer between 1 and 26: ");
    b = Integer.parseInt(cin.nextLine());

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

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

  for (i = a; i <= b; i++) {
    System.out.print(alphabet.charAt(i - 1));
  }
}
```

33. Solution

```
public static void main(String[] args) {
  int attempts, guess, secret_number;

  secret_number = 1 + (int)(Math.random() * 100);

  attempts = 1;
  System.out.print("Enter a guess: ");
  guess = Integer.parseInt(cin.nextLine());
  while (guess != secret_number) {
    if (guess > secret_number) {
      System.out.println("Your guess is bigger than my secret number. Try again.");
    }
    else {
      System.out.println("Your guess is smaller than my secret number. Try again.");
    }
    attempts++;
    System.out.print("Enter a guess: ");
    guess = Integer.parseInt(cin.nextLine());
  }
  System.out.println("You found it!");
  System.out.println("Attempts: " + attempts);
}
```

34. Solution

```
public static void main(String[] args) {
  int attempts = 0, first_player_attempts = 0, guess, i, secret_number;

  for (i = 1; i <= 2; i++) {
    secret_number = 1 + (int)(Math.random() * 100);

    attempts = 1;
    System.out.print("Enter a guess: ");
    guess = Integer.parseInt(cin.nextLine());
    while (guess != secret_number) {
      if (guess > secret_number) {
        System.out.println("Your guess is bigger than my secret number. Try again.");
      }
      else {
        System.out.println("Your guess is smaller than my secret number. Try again.");
      }
      attempts++;
      System.out.print("Enter a guess: ");
      guess = Integer.parseInt(cin.nextLine());
    }
    System.out.println("You found it!");
    System.out.println("Attempts: " + attempts);

    if (i == 1) {
```

```
      first_player_attempts = attempts;
    }
  }

  if (first_player_attempts < attempts) {
    System.out.println("First player wins!");
  }
  else if (first_player_attempts > attempts) {
    System.out.println("Second player wins!");
  }
  else {
    System.out.println("It's a draw");
  }
}
```

35. Solution

```
public static void main(String[] args) {
  int choice, diagonal;

  while (true) {
    System.out.println("1. 4/3 TV Screen");
    System.out.println("2. 16/9 TV Screen");
    System.out.println("3. Exit");
    System.out.print("Enter a choice: ");
    choice = Integer.parseInt(cin.nextLine());

    if (choice == 3) {
      break;
    }
    else if (choice == 1) {
      System.out.println("Enter diagonal: ");
      diagonal = Integer.parseInt(cin.nextLine());
      System.out.println("Width: " + (diagonal * 0.8));
      System.out.println("Height: " + (diagonal * 0.6));
    }
    else if (choice == 2) {
      System.out.println("Enter diagonal: ");
      diagonal = Integer.parseInt(cin.nextLine());
      System.out.println("Width: " + (diagonal * 0.87));
      System.out.println("Height: " + (diagonal * 0.49));
    }
  }
}
```

36. Solution

```
public static void main(String[] args) {
  int count_a, count_a_boys, count_b, count_cdef_girls, grade;
  int i, maximum, minimum, n, total, total_a, total_a_boys, total_b;
  String gender;
```

```
System.out.print("Enter total number of students: ");
n = Integer.parseInt(cin.nextLine());
while (n <= 0) {
  System.out.print("Wrong number. Please enter total number of students: ");
  n = Integer.parseInt(cin.nextLine());
}

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

maximum = -1;
minimum = 101;

for (i = 1; i <= n; i++) {
  System.out.print("Enter grade for student No " + i + ": ");
  grade = Integer.parseInt(cin.nextLine());
  while (grade < 0 || grade > 100) {
    System.out.print("Wrong grade. Please enter grade for student No " + i + ": ");
    grade = Integer.parseInt(cin.nextLine());
  }

  System.out.print("Enter gender for student No " + i + ": ");
  gender = cin.nextLine().toUpperCase();
  while (!gender.equals("M") && !gender.equals("F")) {
    System.out.print("Wrong gender. Please enter gender for student No " + i + ": ");
    gender = cin.nextLine().toUpperCase();
  }

  if (grade >= 90 && grade <= 100) {
    total_a += grade;
    count_a++;
    if (gender.equals("M")) {
      total_a_boys += grade;
      count_a_boys++;
    }
  }
  else if (grade >= 80 && grade <= 89) {
    total_b += grade;
    count_b++;
  }
  else {
    if (gender.equals("F")) {
      count_cdef_girls++;
    }
```

```
    }

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

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

    total += grade;
  }

  if (count_a > 0) {
    System.out.print("The average value of those who got an 'A' is: ");
    System.out.println(total_a / (double)count_a);
  }
  if (count_b > 0) {
    System.out.print("The average value of those who got a 'B' is: ");
    System.out.println(total_b / (double)count_b);
  }
  if (count_a_boys > 0) {
    System.out.print("The average value of boys who got an 'A' is: ");
    System.out.println(total_a_boys / (double)count_a_boys);
  }
  System.out.println("The total number of girls that got less than 'B' is: " + count_cdef_girls);
  System.out.println("The highest grade is: " + maximum);
  System.out.println("The lowest grade is: " + minimum);
  System.out.println("The average grade of the whole class is: " + total / (double)n);
}
```

37. Solution

```
public static void main(String[] args) {
  double amount, discount;
  String answer;

  do {
    System.out.print("Enter amount: ");
    amount = Double.parseDouble(cin.nextLine());
    while (amount <= 0) {
      System.out.println("Wrong amount. Please enter amount: ");
      amount = Double.parseDouble(cin.nextLine());
    }

    if (amount < 20) {
      discount = 0;
    }
    else if (amount < 50) {
      discount = 3;
    }
```

```
    else if (amount < 100) {
      discount = 5;
    }
    else {
      discount = 10;
    }

    System.out.println("Discount: " + discount + "%");
    System.out.println("Amount to pay (discount included): " + (amount - amount * discount / 100));

    System.out.print("Would you like to repeat? ");
    answer = cin.nextLine().toUpperCase();
  } while (answer.equals("YES"));
}
```

38. Solution

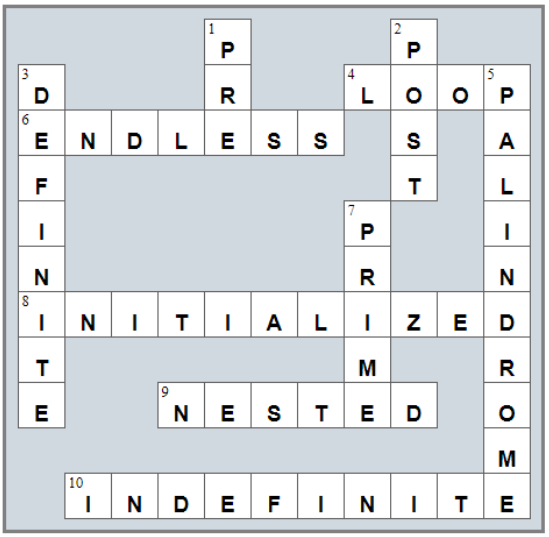
```
static final double TAX_RATE = 0.25;

public static void main(String[] args) {
  int kwh;
  double t;

  System.out.print("Enter number of Kilowatt-hours consumed: ");
  kwh = Integer.parseInt(cin.nextLine());
  while (kwh < 0 && kwh != -1) {
    System.out.print("Wrong value. Please enter number of Kilowatt-hours consumed: ");
    kwh = Integer.parseInt(cin.nextLine());
  }

  while (kwh != -1) {
    if (kwh <= 400) {
      t = kwh * 0.11;
    }
    else if (kwh <= 1500) {
      t = 400 * 0.11 + (kwh - 400) * 0.22;
    }
    else if (kwh <= 3500) {
      t = 400 * 0.11 + 1100 * 0.22 + (kwh - 1500) * 0.25;
    }
    else {
      t = 400 * 0.11 + 1100 * 0.22 + 2000 * 0.25 + (kwh - 3500) * 0.50;
    }

    t += t * TAX_RATE;
    System.out.println("Total amount to pay (taxes included): " + t);

    System.out.print("Enter number of Kilowatt-hours consumed: ");
    kwh = Integer.parseInt(cin.nextLine());
    while (kwh < 0 && kwh != -1) {
      System.out.print("Wrong value. Please enter number of Kilowatt-hours consumed: ");
```

```
      kwh = Integer.parseInt(cin.nextLine());
    }
  }
}
```

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

`Weights =`

170	*0*
190	*1*
193	*2*
165	*3*
200	*4*

People

2. Solution

Names =

John Thompson
Chloe Brown
Ryan Miller
Antony Harris
Alexander Lewis
Samantha Clark
Ava Parker

Weights =

170	*0*
190	*1*
193	*2*
165	*3*
200	*4*
170	*5*
172	*6*

People

3. Solution

Months

Names =

Toba
Issyk Kul
Baikal
Crater
Karakul

Areas =

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

Lakes

June July August

4. Solution

Dimensions

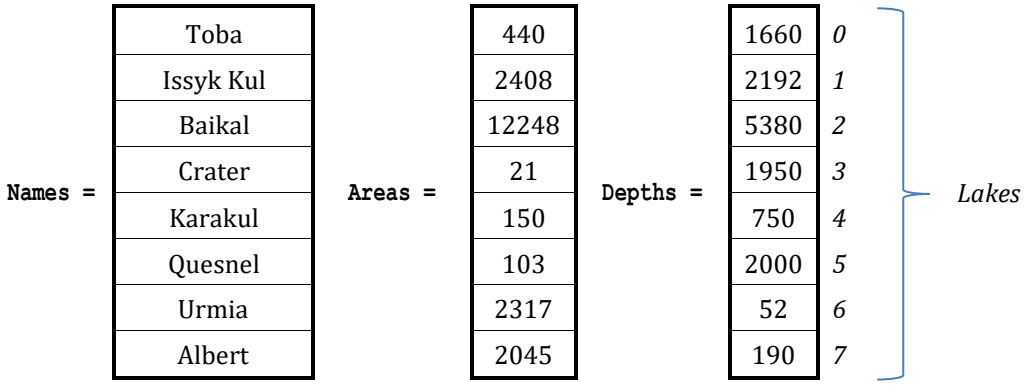
Boxes =

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

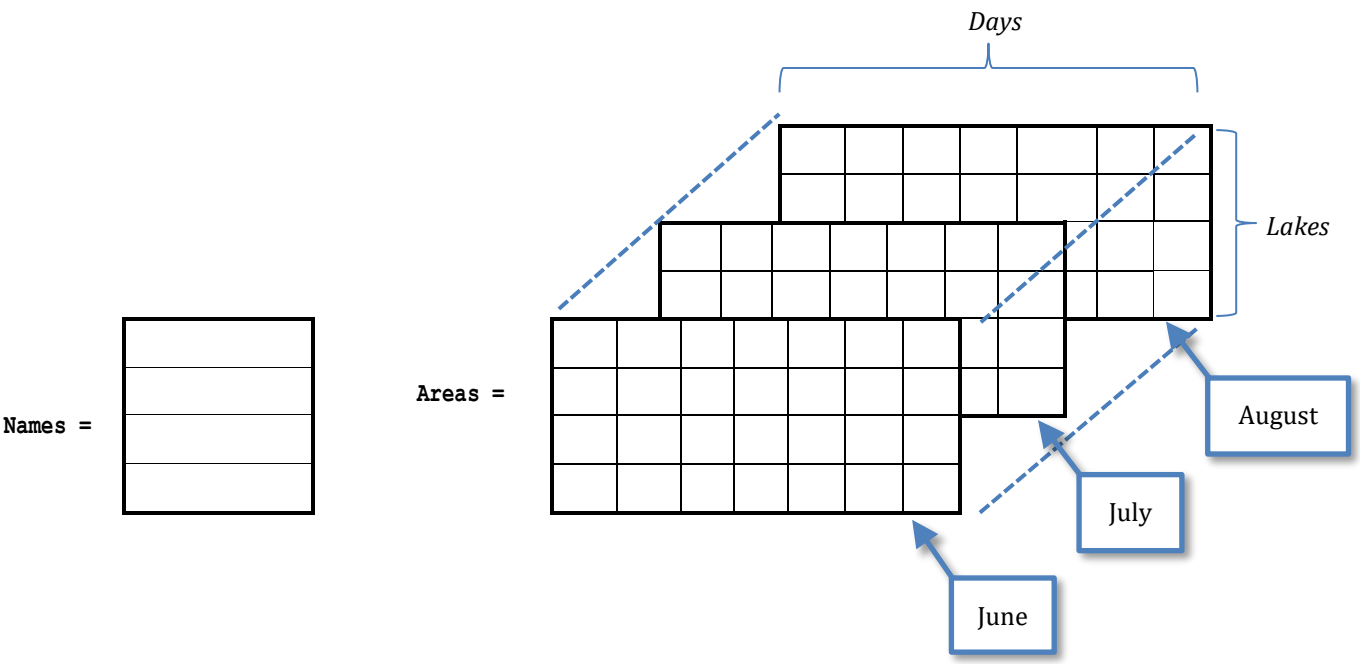
Boxes

Width Height Length

5. Solution



6. Solution



7. Solution

Step	Statement	x	a[0]	a[1]	a[2]
1	`int[] a = new int[3]`	?	?	?	?
2	`a[2] = 1`	?	?	?	**1**
3	`x = 0`	**0**	?	?	1
4	`a[x + a[2]] = 4`	0	?	**4**	1
5	`a[x] = a[x + 1] * 4`	0	**16**	4	1

8. Solution

Step	Statement	x	a[0]	a[1]	a[2]	a[3]	a[4]
1	`int[] a = new int[5]`	?	?	?	?	?	?
2	`a[1] = 5`	?	?	**5**	?	?	?
3	`x = 0`	**0**	?	5	?	?	?
4	`a[x] = 4`	0	**4**	5	?	?	?
5	`a[a[0]] = a[x + 1] % 3`	0	4	5	?	?	**2**
6	`a[a[0] / 2] = 10`	0	4	5	**10**	?	2
7	`x += 2`	**2**	4	5	10	?	2
8	`a[x + 1] = a[x] + 9`	2	4	5	10	**19**	2

9. Solution

For input value of 3

Step	Statement	x	a[0]	a[1]	a[2]	a[3]
1	`int[] a = new int[4]`	?	?	?	?	?
2	`a[1] = Integer.parseInt(cin.nextLine())`	?	?	**3**	?	?
3	`x = 0`	**0**	?	3	?	?
4	`a[x] = 3`	0	**3**	3	?	?
5	`a[a[0]] = a[x + 1] % 2`	0	3	3	?	**1**
6	`a[a[0] % 2] = 10`	0	3	**10**	?	1
7	`x++`	**1**	3	10	?	1
8	`a[x + 1] = a[x] + 9`	1	3	10	**19**	1

For input value of 4

Step	Statement	x	a[0]	a[1]	a[2]	a[3]
1	`int[] a = new int[4]`	?	?	?	?	?
2	`a[1] = Integer.parseInt(cin.nextLine())`	?	?	**4**	?	?
3	`x = 0`	**0**	?	4	?	?
4	`a[x] = 3`	0	**3**	4	?	?
5	`a[a[0]] = a[x + 1] % 2`	0	3	4	?	**0**
6	`a[a[0] % 2] = 10`	0	3	**10**	?	0
7	`x++`	**1**	3	10	?	0
8	`a[x + 1] = a[x] + 9`	1	3	10	**19**	0

For input value of 1

Step	Statement	x	a[0]	a[1]	a[2]	a[3]
1	`int[] a = new int[4]`	?	?	?	?	?
2	`a[1] = Integer.parseInt(cin.nextLine())`	?	?	**1**	?	?
3	`x = 0`	**0**	?	1	?	?

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

10. Solution

For input value of 100

Step	Statement	x	a[0]	a[1]	a[2]	a[3]
1	`int[] a = new int[4]`	?	?	?	?	?
2	`a[1] = Integer.parseInt(cin.nextLine())`	?	?	**100**	?	?
3	`x = 0`	**0**	?	100	?	?
4	`a[x] = 3`	0	**3**	100	?	?
5	`a[a[0]] = a[x + 1] % 10`	0	3	100	?	**0**
6	`if (a[3] > 5)`	false				
7	`a[2] = 3`	0	3	100	**3**	0

For input value of 108

Step	Statement	x	a[0]	a[1]	a[2]	a[3]
1	`int[] a = new int[4]`	?	?	?	?	?
2	`a[1] = Integer.parseInt(cin.nextLine())`	?	?	**108**	?	?
3	`x = 0`	**0**	?	108	?	?
4	`a[x] = 3`	0	**3**	108	?	?
5	`a[a[0]] = a[x + 1] % 10`	0	3	108	?	**8**
6	`if (a[3] > 5)`	true				
7	`a[a[0] % 2] = 9`	0	3	**9**	?	8
8	`x += 1`	**1**	3	9	?	8
9	`a[x + 1] = a[x] + 9`	1	3	9	**18**	8

For input value of 1

Step	Statement	x	a[0]	a[1]	a[2]	a[3]
1	`int[] a = new int[4]`	?	?	?	?	?
2	`a[1] = Integer.parseInt(cin.nextLine())`	?	?	**1**	?	?
3	`x = 0`	**0**	?	1	?	?
4	`a[x] = 3`	0	**3**	1	?	?
5	`a[a[0]] = a[x + 1] % 10`	0	3	1	?	**1**
6	`if (a[3] > 5)`	false				
7	`a[2] = 3`	0	3	1	**3**	1

11. Solution

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

4, 5	`if (x > y)`
	`  a[0] = 1;`
	`else`
	`  a[0] = y;`
		4	3	**1**	?	?

6	`a[1] = x + 3`	4	3	1	**7**	?
7	`y = y - 1`	4	**2**	1	7	?
8	`a[y] = (x + 5) % 2`	4	2	1	7	**1**

12. Solution

Step	Statement	i	a[0]	a[1]	a[2]	a[3]	a[4]	a[5]
1	`int[] a = {17, 12, 45, 12, 12, 49}`	?	**17**	**12**	**45**	**12**	**12**	**49**
2	`i = 0;`	**0**	17	12	45	12	12	49
3	`i <= 5`	true						
4	`if (a[i] == 12)`	false						
5	`a[i]++`	0	**18**	12	45	12	12	49
6	`i++`	**1**	18	12	45	12	12	49
7	`i <= 5`	true						
8	`if (a[i] == 12)`	true						
9	`a[i]--`	1	18	**11**	45	12	12	49
10	`i++`	**2**	18	11	45	12	12	49
11	`i <= 5`	true						
12	`if (a[i] == 12)`	false						
13	`a[i]++`	2	18	11	**46**	12	12	49
14	`i++`	**3**	18	11	46	12	12	49
15	`i <= 5`	true						
16	`if (a[i] == 12)`	true						
17	`a[i]--`	3	18	11	46	**11**	12	49
18	`i++`	**4**	18	11	46	11	12	49
19	`i <= 5`	true						
20	`if (a[i] == 12)`	true						
21	`a[i]--`	4	18	11	46	11	**11**	49
22	`i++`	**5**	18	11	46	11	11	49
23	`i <= 5`	true						

24	`if (a[i] == 12)`	false						
25	`a[i]++`	5	18	11	46	11	11	**50**
26	`i++`	**6**	18	11	46	11	11	50
27	`i <= 5`	false						

13. Solution

Step	Statement	i	a[0]	a[1]	a[2]	a[3]	a[4]	a[5]
1	`int[] a = {10, 15, 12, 23, 22, 19}`	?	**10**	**15**	**12**	**23**	**22**	**19**
2	`i = 1`	**1**	10	15	12	23	22	19
3	`i <= 4`	true						
4	`a[i] = a[i + 1] + a[i - 1]`	1	10	**22**	12	23	22	19
5	`i++`	**2**	10	22	12	23	22	19
6	`i <= 4`	true						
7	`a[i] = a[i + 1] + a[i - 1]`	2	10	22	**45**	23	22	19
8	`i++`	**3**	10	22	45	23	22	19
9	`i <= 4`	true						
10	`a[i] = a[i + 1] + a[i - 1]`	3	10	22	45	**67**	22	19
11	`i++`	**4**	10	22	45	67	22	19
12	`i <= 4`	true						
13	`a[i] = a[i + 1] + a[i - 1]`	4	10	22	45	67	**86**	19
14	`i++`	**5**	10	22	45	67	86	19
15	`i <= 4`	false						

14. Solution

It displays:
Navajo
Cherokee
Sioux

15. Solution

```
static final int ELEMENTS = 100;

public static void main(String[] args) {
  int i;

  double[] a = new double[ELEMENTS];
  for (i = 0; i <= ELEMENTS - 1; i++) {
    a[i] = Double.parseDouble(cin.nextLine());
  }

  for (i = 0; i <= ELEMENTS - 1; i++) {
    System.out.println(Math.pow(a[i], 3));
```

```
  }
}
```

16. Solution

```
static final int ELEMENTS = 80;

public static void main(String[] args) {
  int i;

  double[] a = new double[ELEMENTS];
  for (i = 0; i <= ELEMENTS - 1; i++) {
    a[i] = Double.parseDouble(cin.nextLine());
  }

  for (i = 0; i <= ELEMENTS - 1; i++) {
    a[i] = Math.pow(a[i], 2);
  }

  for (i = ELEMENTS - 1; i >= 0; i--) {
    System.out.println(a[i]);
  }
}
```

17. Solution

```
static final int ELEMENTS = 90;

public static void main(String[] args) {
  int i;

  int[] a = new int[ELEMENTS];
  for (i = 0; i <= ELEMENTS - 1; i++) {
    a[i] = Integer.parseInt(cin.nextLine());
  }

  for (i = ELEMENTS - 1; i >= 0; i--) {
    if (a[i] % 5 == 0) {
      System.out.println(a[i]);
    }
  }
}
```

18. Solution

```
static final int ELEMENTS = 50;

public static void main(String[] args) {
  int i;

  int[] a = new int[ELEMENTS];
  for (i = 0; i <= ELEMENTS - 1; i++) {
    a[i] = Integer.parseInt(cin.nextLine());
```

```
  }

  for (i = 0; i <= ELEMENTS - 1; i++) {
    if (a[i] % 2 == 0 || a[i] > 10) {
      System.out.println(a[i]);
    }
  }
}
```

19. Solution

```
static final int ELEMENTS = 30;

public static void main(String[] args) {
  int i;
  double total;

  double[] a = new double[ELEMENTS];
  for (i = 0; i <= ELEMENTS - 1; i++) {
    a[i] = Double.parseDouble(cin.nextLine());
  }

  total = 0;
  for (i = 0; i <= ELEMENTS - 1; i++) {
    if (a[i] > 0) {
      total += a[i];
    }
  }
  System.out.println(total);
}
```

20. Solution

```
static final int ELEMENTS = 50;

public static void main(String[] args) {
  int i, total;

  int[] a = new int[ELEMENTS];
  for (i = 0; i <= ELEMENTS - 1; i++) {
    a[i] = Integer.parseInt(cin.nextLine());
  }

  total = 0;
  for (i = 0; i <= ELEMENTS - 1; i++) {
    if (a[i] >= 10 && a[i] <= 99) {
      total += a[i];
    }
  }
  System.out.println(total);
}
```

21. Solution

```
static final int ELEMENTS = 40;

public static void main(String[] args) {
  int i;
  double sum_neg, sum_pos;

  double[] a = new double[ELEMENTS];
  for (i = 0; i <= ELEMENTS - 1; i++) {
    a[i] = Double.parseDouble(cin.nextLine());
  }

  sum_pos = 0;
  sum_neg = 0;
  for (i = 0; i <= ELEMENTS - 1; i++) {
    if (a[i] > 0 ) {
      sum_pos += a[i];
    }
    else if (a[i] < 0) {
      sum_neg += a[i];
    }
  }
  System.out.println(sum_pos + ", " + sum_neg);
}
```

22. Solution

```
static final int ELEMENTS = 20;

public static void main(String[] args) {
  int i;
  double total;

  double[] a = new double[ELEMENTS];
  for (i = 0; i <= ELEMENTS - 1; i++) {
    a[i] = Double.parseDouble(cin.nextLine());
  }

  total = 0;
  for (i = 0; i <= ELEMENTS - 1; i++) {
    total += a[i];
  }
  System.out.println(total / ELEMENTS);
}
```

23. Solution

```
static final int ELEMENTS = 50;

public static void main(String[] args) {
  int i;
```

```
  int[] a = new int[ELEMENTS];
  for (i = 0; i <= ELEMENTS - 1; i++) {
    System.out.print("Enter an integer: ");
    a[i] = Integer.parseInt(cin.nextLine());
  }

  for (i = 0; i <= ELEMENTS - 1; i++) {
    if (a[i] < 20) {
      System.out.println(a[i]);
    }
  }
}
```

24. Solution

```
static final int ELEMENTS = 60;

public static void main(String[] args) {
  int i;

  double[] a = new double[ELEMENTS];
  for (i = 0; i <= ELEMENTS - 1; i++) {
    System.out.print("Enter a number: ");
    a[i] = Double.parseDouble(cin.nextLine());
  }

  for (i = 0; i <= ELEMENTS - 1; i += 2) {
    System.out.println(a[i]);
  }
}
```

25. Solution

```
static final int ELEMENTS = 20;

public static void main(String[] args) {
  int i;
  double total;

  double[] a = new double[ELEMENTS];
  for (i = 0; i <= ELEMENTS - 1; i++) {
    System.out.print("Enter a number: ");
    a[i] = Double.parseDouble(cin.nextLine());
  }

  total = 0;
  for (i = 0; i <= ELEMENTS - 1; i += 2) {
    total += a[i];
  }
  System.out.println(total);
}
```

26. Solution

```
static final int ELEMENTS = 100;
```

```
public static void main(String[] args) {
  int i;
  int[] a = new int[ELEMENTS];
  for (i = 0; i <= ELEMENTS - 1; i++) {
    a[i] = i + 1;
  }
…
```

27. Solution

First approach

```
static final int ELEMENTS = 100;

public static void main(String[] args) {
  int i, k;
  int[] a = new int[ELEMENTS];
  k = 2;
  for (i = 0; i <= ELEMENTS - 1; i++) {
    a[i] = k;
    k += 2;
  }
…
```

Second approach

```
static final int ELEMENTS = 100;

public static void main(String[] args) {
  int i;
  int[] a = new int[ELEMENTS];
  for (i = 0; i <= ELEMENTS - 1; i++) {
    a[i] = (i + 1) * 2;
  }
…
```

28. Solution

```
public static void main(String[] args) {
  int i, n;

  System.out.print("Enter N: ");
  n = Integer.parseInt(cin.nextLine());

  int[] a = new int[n];
  for (i = 1; i <= n; i++) {
    a[i - 1] = (int)Math.pow(i, 2);
  }

  for (i = 0; i <= n - 1; i++) {
    System.out.println(a[i]);
  }
}
```

29. Solution

```
static final int ELEMENTS = 10;
```

```
public static void main(String[] args) {
  int i;

  double[] a = new double[ELEMENTS];
  for (i = 1; i <= ELEMENTS - 1; i++) {
    System.out.print("Enter a number: ");
    a[i] = Double.parseDouble(cin.nextLine());
  }

  for (i = 0; i <= ELEMENTS - 1; i++) {
    if (a[i] == (int)a[i]) {
      System.out.println(i);
    }
  }
}
```

30. Solution

```
static final int ELEMENTS = 50;

public static void main(String[] args) {
  int i, count;

  double[] a = new double[ELEMENTS];
  for (i = 1; i <= ELEMENTS - 1; i++) {
    System.out.print("Enter a number: ");
    a[i] = Double.parseDouble(cin.nextLine());
  }

  count = 0;
  for (i = 0; i <= ELEMENTS - 1; i++) {
    if (a[i] < 0) {
      count++;
    }
  }
  System.out.println(count);
}
```

31. Solution

```
static final int WORDS = 50;

public static void main(String[] args) {
  int i;

  String[] a = new String[WORDS];
  for (i = 0; i <= WORDS - 1; i++) {
    a[i] = cin.nextLine();
  }

  for (i = 0; i <= WORDS - 1; i++) {
    if (a[i].length() >= 10 ) {
```

```
      System.out.println(a[i]);
    }
  }
}
```

32. Solution

```
static final int ELEMENTS = 30;

public static void main(String[] args) {
  int i, k;

  String[] words = new String[ELEMENTS];
  for (i = 0; i <= ELEMENTS - 1; i++) {
    words[i] = cin.nextLine();
  }

  int[] length_limits = {0, 5, 10, 20};

  for (k = 1; k <= 3; k++) {
    for (i = 0; i <= ELEMENTS - 1; i++) {
      if (words[i].length() >= length_limits[k - 1] && words[i].length() < length_limits[k]) {
        System.out.println(words[i]);
      }
    }
  }
}
```

33. Solution

```
static final int WORDS = 40;

public static void main(String[] args) {
  int count, i, j;

  String[] a = new String[WORDS];
  for (i = 0; i <= WORDS - 1; i++) {
    System.out.print("Enter a word: ");
    a[i] = cin.nextLine();
  }

  for (i = 0; i <= WORDS - 1; i++) {
    count = 0;
    for (j = 0; j <= a[i].length() - 1; j++) {
      if (a[i].substring(j, j + 1).equals("w")) {
        count++;
      }
    }
    if (count >= 2) {
      System.out.println(a[i]);
    }
  }
}
```

Chapter 32

32.7 Review Questions: True/False

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

32.8 Review Questions: Multiple Choice

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

32.9 Review Exercises

1. Solution

Step	Statement	x	a
1	`int[][] a = new int[2][3]`	?	? ? ? ? ? ?
2	`a[0][2] = 1`	?	? ? **1** ? ? ?
3	`x = 0`	**0**	? ? 1 ? ? ?
4	`a[0][x] = 9`	0	**9** ? 1 ? ? ?
5	`a[0][x + a[0][2]] = 4`	0	9 **4** 1 ? ? ?
6	`a[a[0][2]][2] = 19`	0	9 4 1 ? ? **19**
7	`a[a[0][2]][x + 1] = 13`	0	9 4 1 ? **13** 19
8	`a[a[0][2]][x] = 15`	0	9 4 1 **15** 13 19

2. Solution

Step	Statement	i	j	a
1	`int[][] a = new int[2][3]`	?	?	? ? ? ? ? ?
2	`i = 0`	**0**	?	? ? ? ? ? ?
3	`i <= 1`	true		
4	`j = 0`	0	**0**	? ? ? ? ? ?

5	`j <= 2`			true
6	`a[i][j] = (i + 1) * 5 + j`	0	0	**5** ? ? ? ? ?
7	`j++`	0	**1**	5 ? ? ? ? ?
8	`j <= 2`			true
9	`a[i][j] = (i + 1) * 5 + j`	0	1	5 **6** ? ? ? ?
10	`j++`	0	**2**	5 6 ? ? ? ?
11	`j <= 2`			true
12	`a[i][j] = (i + 1) * 5 + j`	0	2	5 6 **7** ? ? ?
13	`j++`	0	**3**	5 6 7 ? ? ?
14	`j <= 2`			false
15	`i++`	**1**	3	5 6 7 ? ? ?
16	`i <= 1`			true
17	`j = 0`	1	**0**	5 6 7 ? ? ?
18	`j <= 2`			true
19	`a[i][j] = (i + 1) * 5 + j`	1	0	5 6 7 **10** ? ?
20	`j++`	1	**1**	5 6 7 10 ? ?
21	`j <= 2`			true
22	`a[i][j] = (i + 1) * 5 + j`	1	1	5 6 7 10 **11** ?
23	`j++`	1	**2**	5 6 7 10 11 ?

24	`j <= 2`	true		
25	`a[i][j] = (i + 1) * 5 + j`	1	2	5 6 7 10 11 **12**
26	`j++`	1	**3**	5 6 7 10 11 12
27	`j <= 2`	false		
28	`i++`	**2**	3	5 6 7 10 11 12
29	`i <= 1`	false		

3. Solution

Step	Statement	i	j	a
1	`int[][] a = new int[3][3]`	?	?	? ? ? ? ? ? ? ? ?
2	`j = 0`	?	**0**	? ? ? ? ? ? ? ? ?
3	`j <= 2`	True		
4	`i = 0`	0	**0**	? ? ? ? ? ? ? ? ?
5	`i <= 2`	True		
6	`a[i][j] = (i + 1) * 2 + j * 4`	0	0	**2** ? ? ? ? ? ? ? ?
7	`i++`	**1**	0	2 ? ? ? ? ? ? ? ?
8	`i <= 2`	True		
9	`a[i][j] = (i + 1) * 2 + j * 4`	1	0	2 ? ? **4** ? ? ? ? ?

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

26	`i++`	**3**	1	2 6 ? 4 8 ? 6 10 ?
27	`i <= 2`			False
28	`j++`	3	**2**	2 6 ? 4 8 ? 6 10 ?
29	`j <= 2`			True
30	`i = 0`	**0**	2	2 6 ? 4 8 ? 6 10 ?
31	`i <= 2`			True
32	`a[i][j] = (i + 1) * 2 + j * 4`	0	2	2 6 **10** 4 8 ? 6 10 ?
33	`i++`	**1**	2	2 6 10 4 8 ? 6 10 ?
34	`i <= 2`			True
35	`a[i][j] = (i + 1) * 2 + j * 4`	1	2	2 6 10 4 8 **12** 6 10 ?
36	`i++`	**2**	2	2 6 10 4 8 12 6 10 ?
37	`i <= 2`			True
38	`a[i][j] = (i + 1) * 2 + j * 4`	2	2	2 6 10 4 8 12 6 10 **14**
39	`i++`	**3**	2	2 6 10 4 8 12 6 10 14
40	`i <= 2`			False
41	`j++`	3	**3**	2 6 10 4 8 12 6 10 14

42	`j <= 2`	False

4. Solution

For input value of 5

0	5	10
0	6	12

For input value of 9

0	9	18
0	10	20

For input value of 3

0	3	6
0	4	8

5. Solution

For input value of 13

0	3	3
0	17	18

For input value of 10

0	10	3
0	11	15

For input value of 8

3	3	3
11	12	13

6. Solution

19	5	31
28	6	20

7. Solution

26	29
37	34
59	49

8. Solution

i. −1 15 22 25 12 16 7 9 1
ii. 7 9 1 25 12 16 −1 15 22
iii. 22 15 −1 16 12 25 1 9 7

iv. 1 9 7 16 12 25 22 15 −1
v. −1 25 7 15 12 9 22 16 1
vi. 7 25 −1 9 12 15 1 16 22
vii. 22 16 1 15 12 9 −1 25 7
viii. 1 16 22 9 12 15 7 25 −1

9. Solution

```
static final int ROWS = 10;
static final int COLUMNS = 15;

public static void main(String[] args) {
  int i, j;

  int[][] a = new int[ROWS][COLUMNS];
  for (i = 0; i <= ROWS - 1; i++) {
    for (j = 0; j <= COLUMNS - 1; j++) {
      a[i][j] = Integer.parseInt(cin.nextLine());
    }
  }

  for (i = 0; i <= ROWS - 1; i++) {
    for (j = 0; j <= COLUMNS - 1; j++) {
      if (a[i][j] % 2 != 0) {
        System.out.println(i + ", " + j);
      }
    }
  }
}
```

10. Solution

```
static final int ROWS = 10;
static final int COLUMNS = 6;

public static void main(String[] args) {
  int i, j;

  double[][] a = new double[ROWS][COLUMNS];
  for (i = 0; i <= ROWS - 1; i++) {
    for (j = 0; j <= COLUMNS - 1; j++) {
      a[i][j] = Double.parseDouble(cin.nextLine());
    }
  }

  for (i = 0; i <= ROWS - 1; i++) {
    for (j = 0; j <= COLUMNS - 1; j += 2) {
      System.out.println(a[i][j]);
    }
  }
}
```

11. Solution

```
static final int ROWS = 12;
static final int COLUMNS = 8;

public static void main(String[] args) {
  int i, j;
  double total;

  double[][] a = new double[ROWS][COLUMNS];
  for (i = 0; i <= ROWS - 1; i++) {
    for (j = 0; j <= COLUMNS - 1; j++) {
      a[i][j] = Double.parseDouble(cin.nextLine());
    }
  }

  total = 0;
  for (i = 1; i <= ROWS - 1; i += 2) {
    for (j = 0; j <= COLUMNS - 1; j += 2) {
      total += a[i][j];
    }
  }
  System.out.println(total);
}
```

12. Solution

```
static final int N = 8 ;

public static void main(String[] args) {
  int i, j, k;
  double sum_antidiagonal, sum_diagonal;

  double[][] a = new double[N][N];
  for (i = 0; i <= N - 1; i++) {
    for (j = 0; j <= N - 1; j++) {
      a[i][j] = Double.parseDouble(cin.nextLine());
    }
  }

  sum_diagonal = 0;
  sum_antidiagonal = 0;
  for (k = 0; k <= N - 1; k++) {
    sum_diagonal += a[k][k];
    sum_antidiagonal += a[k][N - k - 1];
  }
  System.out.println(sum_diagonal / N + ", " + sum_antidiagonal / N);
}
```

13. Solution

```
static final int N = 5;

public static void main(String[] args) {
  int i, j;

  int[][] a = new int[N][N];
  for (i = 0; i <= N - 1; i++) {
    for (j = 0; j <= N - 1; j++) {
      if (i == N - j - 1) {
        a[i][j] = 5;
      }
      else if (i > N - j - 1) {
        a[i][j] = 88;
      }
      else {
        a[i][j] = 11;
      }
    }
  }

  for (i = 0; i <= N - 1; i++) {
    for (j = 0; j <= N - 1; j++) {
      System.out.print(a[i][j] + "\t");
    }
    System.out.println();
  }
}
```

14. Solution

```
static final int N = 5;

public static void main(String[] args) {
  int i, j;

  int[][] a = new int[N][N];
  for (i = 0; i <= N - 1; i++) {
    for (j = 0; j <= N - 1; j++) {
      if (i == N - j - 1) {
        a[i][j] = 5;
      }
      else if (i > N - j - 1) {
        a[i][j] = 88;
      }
      else {
        a[i][j] = 11;
      }
      if (i == j) {
        a[i][j] = 0;
```

```
      }
    }
  }

  for (i = 0; i <= N - 1; i++) {
    for (j = 0; j <= N - 1; j++) {
      System.out.print(a[i][j] + "\t");
    }
    System.out.println();
  }
}
```

15. Solution

```
static final int ROWS = 5;
static final int COLUMNS = 4;

public static void main(String[] args) {
  int i, j;

  double[][] a = new double[ROWS][COLUMNS];
  for (i = 0; i <= ROWS - 1; i++) {
    for (j = 0; j <= COLUMNS - 1; j++) {
      a[i][j] = Double.parseDouble(cin.nextLine());
    }
  }

  for (i = 0; i <= ROWS - 1; i++) {
    for (j = 0; j <= COLUMNS - 1; j++) {
      if (a[i][j] == (int)(a[i][j])) {
        System.out.println(i + ", " + j);
      }
    }
  }
}
```

16. Solution

```
static final int ROWS = 10;
static final int COLUMNS = 4;

public static void main(String[] args) {
  int count, i, j;

  double[][] a = new double[ROWS][COLUMNS];
  for (i = 0; i <= ROWS - 1; i++) {
    for (j = 0; j <= COLUMNS - 1; j++) {
      a[i][j] = Double.parseDouble(cin.nextLine());
    }
  }
```

```
  count = 0;
  for (i = 0; i <= ROWS - 1; i++) {
    for (j = 0; j <= COLUMNS - 1; j++) {
      if (a[i][j] < 0) {
        count++;
      }
    }
  }
  System.out.println(count);
}
```

17. Solution

```
static final int ROWS = 3;
static final int COLUMNS = 4;

public static void main(String[] args) {
  int i, j;

  String[][] a = new String[ROWS][COLUMNS];
  for (i = 0; i <= ROWS - 1; i++) {
    for (j = 0; j <= COLUMNS - 1; j++) {
      a[i][j] = cin.nextLine();
    }
  }

  for (i = 0; i <= ROWS - 1; i++) {
    for (j = 0; j <= COLUMNS - 1; j++) {
      System.out.print(a[i][j] + " ");
    }
  }
}
```

18. Solution

```
static final int ROWS = 20;
static final int COLUMNS = 14;

public static void main(String[] args) {
  int i, j;

  String[][] a = new String[ROWS][COLUMNS];
  for (i = 0; i <= ROWS - 1; i++) {
    for (j = 0; j <= COLUMNS - 1; j++) {
      a[i][j] = cin.nextLine();
    }
  }

  for (i = 0; i <= ROWS - 1; i++) {
    for (j = 0; j <= COLUMNS - 1; j++) {
      if (a[i][j].length() < 5) {
```

```
        System.out.println(a[i][j]);
      }
    }
  }
}
```

19. Solution

First approach

```
static final int ROWS = 20;
static final int COLUMNS = 14;

public static void main(String[] args) {
  int i, j, k;

  String[][] a = new String[ROWS][COLUMNS];
  for (i = 0; i <= ROWS - 1; i++) {
    for (j = 0; j <= COLUMNS - 1; j++) {
      a[i][j] = cin.nextLine();
    }
  }

  int[] length_limits = {5, 10, 20};

  for (k = 0; k <= 2; k++) {
    for (i = 0; i <= ROWS - 1; i++) {
      for (j = 0; j <= COLUMNS - 1; j++) {
        if (a[i][j].length() < length_limits[k]) {
          System.out.println(a[i][j]);
        }
      }
    }
  }
}
```

Second approach

```
static final int ROWS = 20;
static final int COLUMNS = 14;

public static void main(String[] args) {
  int i, j, k;

  String[][] a = new String[ROWS][ COLUMNS];
  for (i = 0; i <= ROWS - 1; i++) {
    for (j = 0; j <= COLUMNS - 1; j++) {
      a[i][j] = cin.nextLine();
    }
  }

  for (k = 0; k <= 2; k++) {
    for (i = 0; i <= ROWS - 1; i++) {
      for (j = 0; j <= COLUMNS - 1; j++) {
```

```
        if (a[i][j].length() < 5 * Math.pow(2, k)) {
          System.out.println(a[i][j]);
        }
      }
    }
  }
}
```

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

```
static final int STUDENTS = 15;
static final int TESTS = 5;

public static void main(String[] args) {
  int i, j;

  int[][] grades = new int[STUDENTS][TESTS];
  for (i = 0; i <= STUDENTS - 1; i++) {
    for (j = 0; j <= TESTS - 1; j++) {
      grades[i][j] = Integer.parseInt(cin.nextLine());
    }
  }

  double[] average = new double[STUDENTS];
  for (i = 0; i <= STUDENTS - 1; i++) {
    average[i] = 0;
    for (j = 0; j <= TESTS - 1; j++) {
      average[i] += grades[i][j];
    }
    average[i] /= TESTS;
  }

  for (i = 0; i <= STUDENTS - 1; i++) {
    System.out.print("Student No " + (i + 1) + ": ");

    if (average[i] < 60) {
      System.out.println("E/F");
    }
    else if (average[i] < 70) {
      System.out.println("D");
    }
```

```
    else if (average[i] < 80) {
      System.out.println("C");
    }
    else if (average[i] < 90) {
      System.out.println("B");
    }
    else {
      System.out.println("A");
    }
  }
}
```

2. Solution

```
static final int OBJECTS = 5;
static final int FALLS = 10;

public static void main(String[] args) {
  int i, j, total;

  int[][] g = new int[OBJECTS][FALLS];
  for (i = 0; i <= OBJECTS - 1; i++) {
    for (j = 0; j <= FALLS - 1; j++) {
      g[i][j] = Integer.parseInt(cin.nextLine());
    }
  }

  for (i = 0; i <= OBJECTS - 1; i++) {
    total = 0;
    for (j = 0; j <= FALLS - 1; j++) {
      total += g[i][j];
    }
    System.out.println("Average g for object No " + (i + 1) + ": " + (total / (double)FALLS));
  }

  for (j = 0; j <= FALLS - 1; j++) {
    total = 0;
    for (i = 0; i <= OBJECTS - 1; i++) {
      total += g[i][j];
    }
    System.out.println("Average g for fall No " + (j + 1) + ": " + (total / (double)OBJECTS));
  }

  total = 0;
  for (i = 0; i <= OBJECTS - 1; i++) {
    for (j = 0; j <= FALLS - 1; j++) {
      total += g[i][j];
    }
  }
  System.out.println("Overall average g: " + (total / (double)(OBJECTS * FALLS)));
}
```

3. Solution

```
static final int PLAYERS = 15;
static final int MATCHES = 12;

public static void main(String[] args) {
  int i, j, total;

  int[][] points = new int[PLAYERS][MATCHES];
  for (i = 0; i <= PLAYERS - 1; i++) {
    for (j = 0; j <= MATCHES - 1; j++) {
      points[i][j] = Integer.parseInt(cin.nextLine());
    }
  }

  for (i = 0; i <= PLAYERS - 1; i++) {
    total = 0;
    for (j = 0; j <= MATCHES - 1; j++) {
      total += points[i][j];
    }
    System.out.println("Total number of points for player No " +(i + 1) + ": " + total);
  }

  for (j = 0; j <= MATCHES - 1; j++) {
    total = 0;
    for (i = 0; i <= PLAYERS - 1; i++) {
      total += points[i][j];
    }
    System.out.println("Total number of points for match No " + (j + 1) + ": " + total);
  }
}
```

4. Solution

```
static final int CITIES = 20;
static final int HOURS = 24;

public static void main(String[] args) {
  int i, j;
  double total;

  double[][] temperatures = new double[CITIES][HOURS];
  for (i = 0; i <= CITIES - 1; i++) {
    for (j = 0; j <= HOURS - 1; j++) {
      temperatures[i][j] = Integer.parseInt(cin.nextLine());
    }
  }

  for (j = 0; j <= HOURS - 1; j++) {
    total = 0;
    for (i = 0; i <= CITIES - 1; i++) {
```

```
      total += temperatures[i][j];
    }
    if (total / CITIES < 10) {
      System.out.println("Hour: " + (j + 1));
    }
  }
}
```

5. Solution

```
static final int PLAYERS = 24;
static final int MATCHES = 10;

public static void main(String[] args) {
  int i, j, total;

  String[] names = new String[PLAYERS];
  int[][] goals = new int[PLAYERS][MATCHES];
  for (i = 0; i <= PLAYERS - 1; i++) {
    names[i] = cin.nextLine();
    for (j = 0; j <= MATCHES - 1; j++) {
      goals[i][j] = Integer.parseInt(cin.nextLine());
    }
  }

  for (i = 0; i <= PLAYERS - 1; i++) {
    total = 0;
    for (j = 0; j <= MATCHES - 1; j++) {
      total += goals[i][j];
    }
    System.out.println(names[i] + ": " + (total / (double)MATCHES));
  }

  for (j = 0; j <= MATCHES - 1; j++) {
    total = 0;
    for (i = 0; i <= PLAYERS - 1; i++) {
      total += goals[i][j];
    }
    System.out.println("Match No " + (j + 1) + ": " + total);
  }
}
```

6. Solution

```
static final int STUDENTS = 12;
static final int LESSONS = 6;

public static void main(String[] args) {
  int i, j, total;

  String[] names = new String[STUDENTS];
```

```
  int[][] grades = new int[STUDENTS][LESSONS];
  for (i = 0; i <= STUDENTS - 1; i++) {
    names[i] = cin.nextLine();
    for (j = 0; j <= LESSONS - 1; j++) {
      grades[i][j] = Integer.parseInt(cin.nextLine());
    }
  }

  double[] average = new double[STUDENTS];
  for (i = 0; i <= STUDENTS - 1; i++) {
    total = 0;
    for (j = 0; j <= LESSONS - 1; j++) {
      total += grades[i][j];
    }
    average[i] = total / (double)LESSONS;
    System.out.println(names[i] + ": " + average[i]);
  }

  for (j = 0; j <= LESSONS - 1; j++) {
    total = 0;
    for (i = 0; i <= STUDENTS - 1; i++) {
      total += grades[i][j];
    }
    System.out.println(total / (double)STUDENTS);
  }

  for (i = 0; i <= STUDENTS - 1; i++) {
    if (average[i] < 60) {
      System.out.println(names[i]);
    }
  }

  for (i = 0; i <= STUDENTS - 1; i++) {
    if (average[i] > 89) {
      System.out.println(names[i] + " Bravo!");
    }
  }
}
```

7. Solution

```
static final int ARTISTS = 15;
static final int JUDGES = 5;

public static void main(String[] args) {
  int i, j, total;

  String[] judge_names = new String[JUDGES];
  for (j = 0; j <= JUDGES - 1; j++) {
    System.out.print("Enter name for judge No " + (j + 1) + ": ");
    judge_names[j] = cin.nextLine();
```

```
  }

  String[] artist_names = new String[ARTISTS];
  String[] song_titles = new String[ARTISTS];
  int[][] score = new int[ARTISTS][JUDGES];
  for (i = 0; i <= ARTISTS - 1; i++) {
    System.out.print("Enter name for artist No " + (i + 1) + ": ");
    artist_names[i] = cin.nextLine();
    System.out.print("Enter song title for artist " + artist_names[i] + ": ");
    song_titles[i] = cin.nextLine();
    for (j = 0; j <= JUDGES - 1; j++) {
      System.out.print("Enter score for artist: " + artist_names[i]);
      System.out.print(" gotten from judge " + judge_names[j] + ": ");
      score[i][j] = Integer.parseInt(cin.nextLine());
    }
  }

  for (i = 0; i <= ARTISTS - 1; i++) {
    total = 0;
    for (j = 0; j <= JUDGES - 1; j++) {
      total += score[i][j];
    }
    System.out.println(artist_names[i] + ", " + song_titles[i] + ": " + total);
  }

  for (j = 0; j <= JUDGES - 1; j++) {
    total = 0;
    for (i = 0; i <= ARTISTS - 1; i++) {
      total += score[i][j];
    }
    System.out.println(judge_names[j] + ": " + total / (double)ARTISTS);
  }
}
```

8. Solution

```
static final int PEOPLE = 30;
static final int MONTHS = 12;

public static void main(String[] args) {
  int i, j, sum_heights, sum_weights;
  double average_height, average_weight;

  int[][] weights = new int[PEOPLE][MONTHS];
  int[][] heights = new int[PEOPLE][MONTHS];
  for (i = 0; i <= PEOPLE - 1; i++) {
    for (j = 0; j <= MONTHS - 1; j++) {
      weights[i][j] = Integer.parseInt(cin.nextLine());
      heights[i][j] = Integer.parseInt(cin.nextLine());
    }
  }
```

```
  for (i = 0; i <= PEOPLE - 1; i++) {
    sum_weights = 0;
    sum_heights = 0;
    for (j = 0; j <= MONTHS - 1; j++) {
      sum_weights += weights[i][j];
      sum_heights += heights[i][j];
    }
    average_weight = sum_weights / (double)MONTHS;
    average_height = sum_heights / (double)MONTHS;
    System.out.println(average_weight + ", " + average_height);
    System.out.println(average_weight * 702 / Math.pow(average_height, 2));
  }

  for (i = 0; i <= PEOPLE - 1; i++) {
    System.out.println(weights[i][4] * 702 / Math.pow(heights[i][4], 2));
    System.out.println(weights[i][7] * 702 / Math.pow(heights[i][7], 2));
  }
}
```

9. Solution

```
static final double VAT = 0.19;
static final int CONSUMERS = 1000;

public static void main(String[] args) {
  int consumed, i;
  double payment, total;

  int[][] meter_reading = new int[CONSUMERS][2];
  for (i = 0; i <= CONSUMERS - 1; i++) {
    meter_reading[i][0] = Integer.parseInt(cin.nextLine());
    meter_reading[i][1] = Integer.parseInt(cin.nextLine());
  }

  total = 0;
  for (i = 0; i <= CONSUMERS - 1; i++) {
    consumed = meter_reading[i][1] - meter_reading[i][0];
    System.out.println(consumed);
    payment = consumed * 0.07;
    payment += VAT * payment;
    System.out.println(payment);

    total += consumed;
  }

  System.out.println(total + ", " + (total * 0.07 + total * 0.07 * VAT));
}
```

10. Solution

```
static final int CURRENCIES = 4;
static final int DAYS = 5;

public static void main(String[] args) {
  int i, j;
  double average, total, usd;

  System.out.print("Enter an amount in US dollars: ");
  usd = Double.parseDouble(cin.nextLine());

  String[] currency = {"British Pounds Sterling", "Euros", "Canadian Dollars", "Australian Dollars"};

  double[][] rate = {{1.320, 1.321, 1.332, 1.331, 1.341},
                     {1.143, 1.156, 1.138, 1.122, 1.129},
                     {0.757, 0.764, 0.760, 0.750, 0.749},
                     {0.720, 0.725, 0.729, 0.736, 0.739}
                    };

  for (i = 0; i <= CURRENCIES - 1; i++) {
    total = 0;
    for (j = 0; j <= DAYS - 1; j++) {
      total += rate[i][j];
    }
    average = total / DAYS;
    System.out.println(usd + " US dollars = " + (usd / average) + " " + currency[i]);
  }
}
```

11. Solution

```
static final int EMPLOYEES = 10;
static final int DAYS = 5;

public static void main(String[] args) {
  int i, j;
  double gross_pay, pay_rate, total;

  String[] days = {"Monday", "Tuesday", "Wednesday", "Thursday", "Friday"};

  pay_rate = Double.parseDouble(cin.nextLine());

  String[] names = new String[EMPLOYEES];
  int[][] hours_worked_per_day = new int[EMPLOYEES][DAYS];
  for (i = 0; i <= EMPLOYEES - 1; i++) {
    names[i] = cin.nextLine();
    for (j = 0; j <= DAYS - 1; j++) {
      hours_worked_per_day[i][j] = Integer.parseInt(cin.nextLine());
    }
```

```
  }

  int[] hours_worked_per_week = new int[EMPLOYEES];
  for (i = 0; i <= EMPLOYEES - 1; i++) {
    hours_worked_per_week[i] = 0;
    for (j = 0; j <= DAYS - 1; j++) {
      hours_worked_per_week[i] += hours_worked_per_day[i][j];
    }
    if (hours_worked_per_week[i] > 40) {
      System.out.println(names[i]);
    }
  }

  for (i = 0; i <= EMPLOYEES - 1; i++) {
    if (hours_worked_per_week[i] <= 40) {
      gross_pay = pay_rate * hours_worked_per_week[i];
    }
    else {
      gross_pay = pay_rate * 40 + 1.5 * pay_rate * (hours_worked_per_week[i] - 40);
    }
    System.out.println(names[i] + ", " + gross_pay);
  }

  for (i = 0; i <= EMPLOYEES - 1; i++) {
    if (hours_worked_per_week[i] > 40) {
      for (j = 0; j <= DAYS - 1; j++) {
        if (hours_worked_per_day[i][j] > 8) {
          System.out.println(names[i] + ", " + days[j] + " Overtime!");
        }
      }
    }
  }

  for (j = 0; j <= DAYS - 1; j++) {
    total = 0;
    for (i = 0; i <= EMPLOYEES - 1; i++) {
      if (hours_worked_per_day[i][j] <= 8) {
        gross_pay = pay_rate * hours_worked_per_day[i][j];
      }
      else {
        gross_pay = pay_rate * 8 + 1.5 * pay_rate * (hours_worked_per_day[i][j] - 8);
      }
      total += gross_pay;
    }
    System.out.println(days[j] + ", " + total);
  }
}
```

12. Solution

```
static final int ROWS = 3;
```

```
static final int COLUMNS = 4;

public static void main(String[] args) {
  int i, j, k;

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

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

  for (k = 0; k <= b.length - 1; k++) {
    System.out.print(b[k] + " ");
  }
}
```

13. Solution

```
static final int ROWS = 3;
static final int COLUMNS = 3;

public static void main(String[] args) {
  int i, j, k;

  int[] a = {16, 12, 3, 5, 6, 9, 18, 19, 20};

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

  for (i = 0; i <= ROWS - 1; i++) {
    for (j = 0; j <= COLUMNS - 1; j++) {
      System.out.print(b[i][j] + "\t");
    }
    System.out.println();
  }
}
```

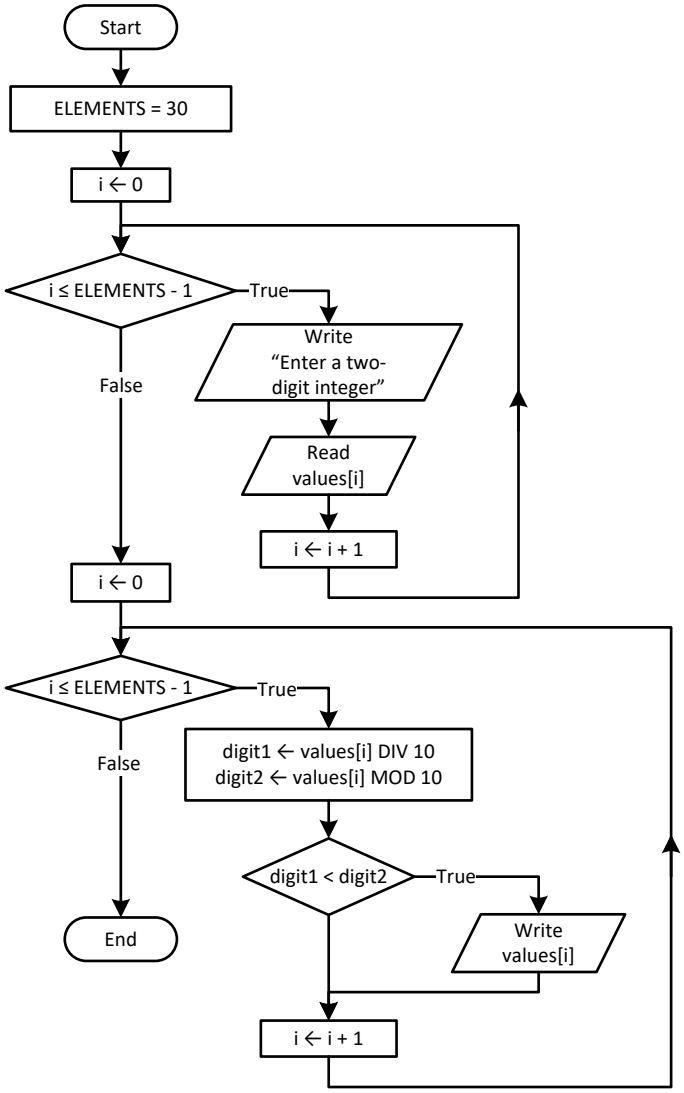
Chapter 34

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

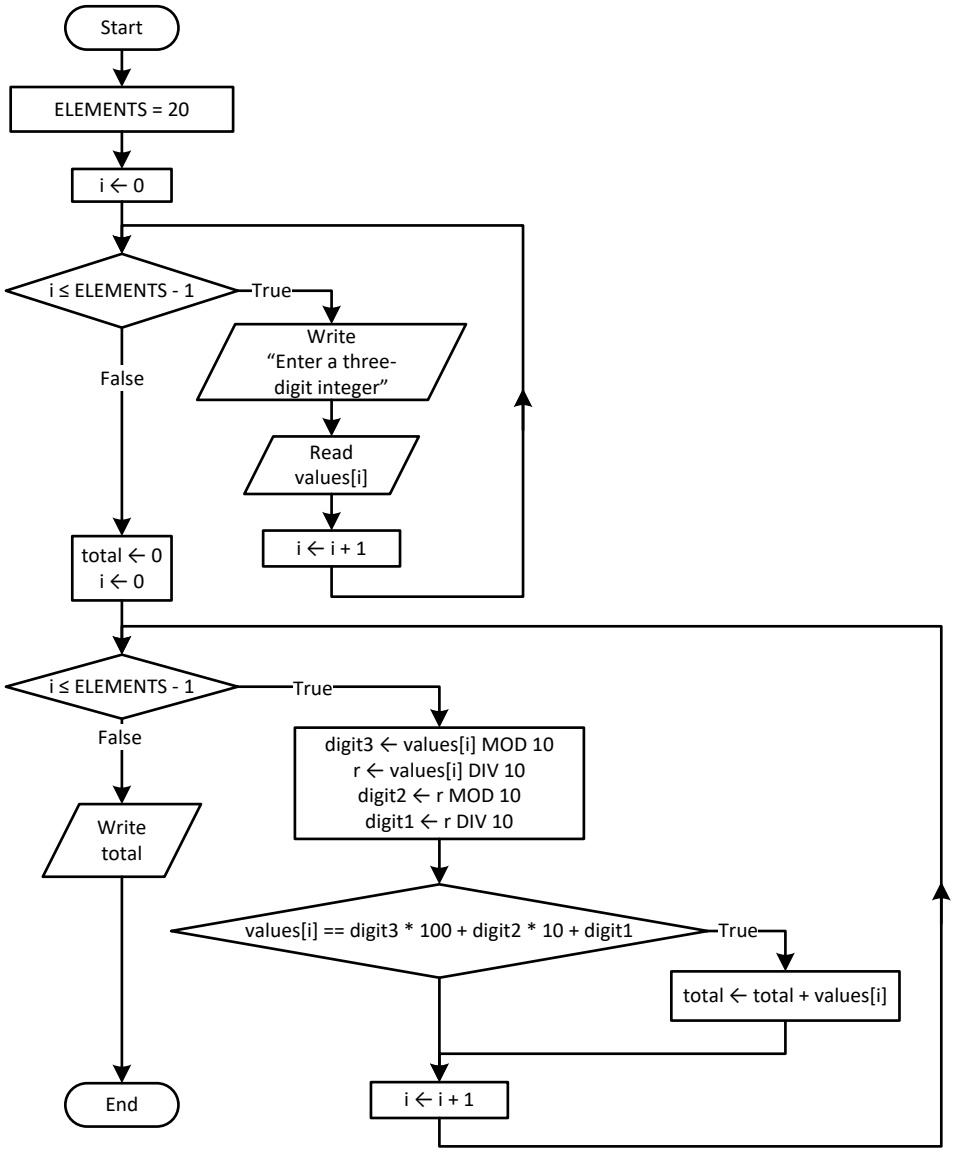
34.8 Review Exercises

1. Solution

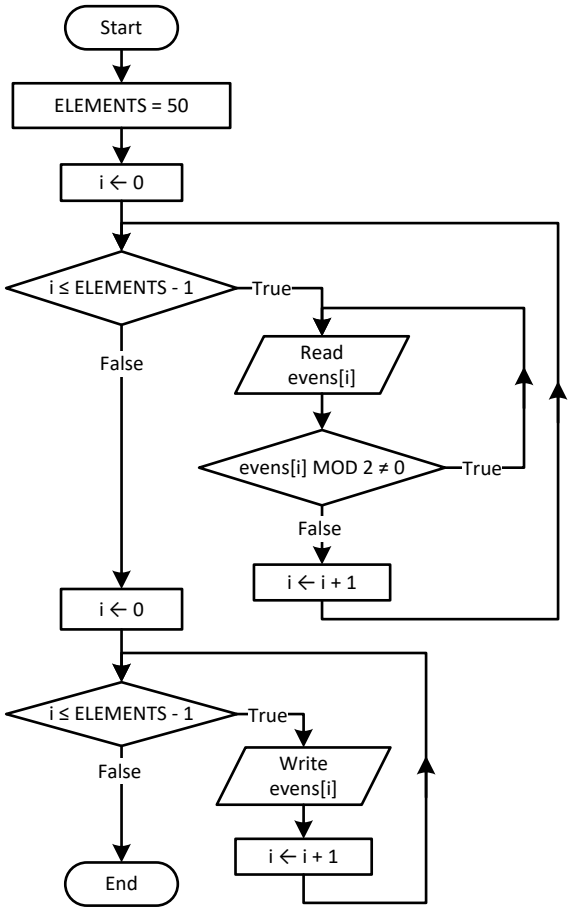


✎ *Please note that since flowcharts are a loose method to represent an algorithm, it is not necessary to initialize an array within a flowchart; that is, there is no need to represent the statement* `int[] values = new int[ELEMENTS]`.

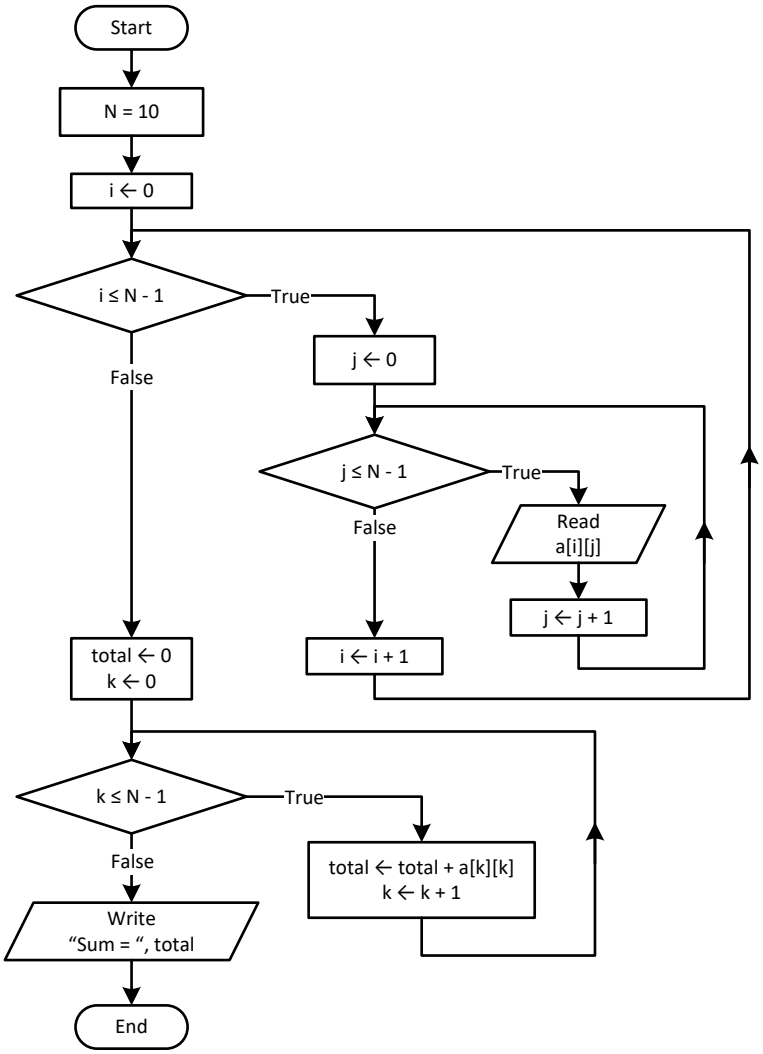
2. Solution


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

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; i <= CITIES - 1; i++) {
  do {
    b[i] = Double.parseDouble(cin.nextLine());
  } while (b[i] >= 0);
}
```

6. Solution

```
public static void main(String[] args) {
  int i, m, n;
  double b;
  double[] pos = new double[90];
  double[] neg = new double[90];

  i = 1;
  m = 0;
```

```
  n = 0;
  do {
    b = Double.parseDouble(cin.nextLine());
    if (b < 0) {
      pos[m] = b;
      m++;
    }
    else {
      neg[n] = b;
      n++;
    }
    i++;
  } while (i < 90);
  System.out.println("The End");
}
```

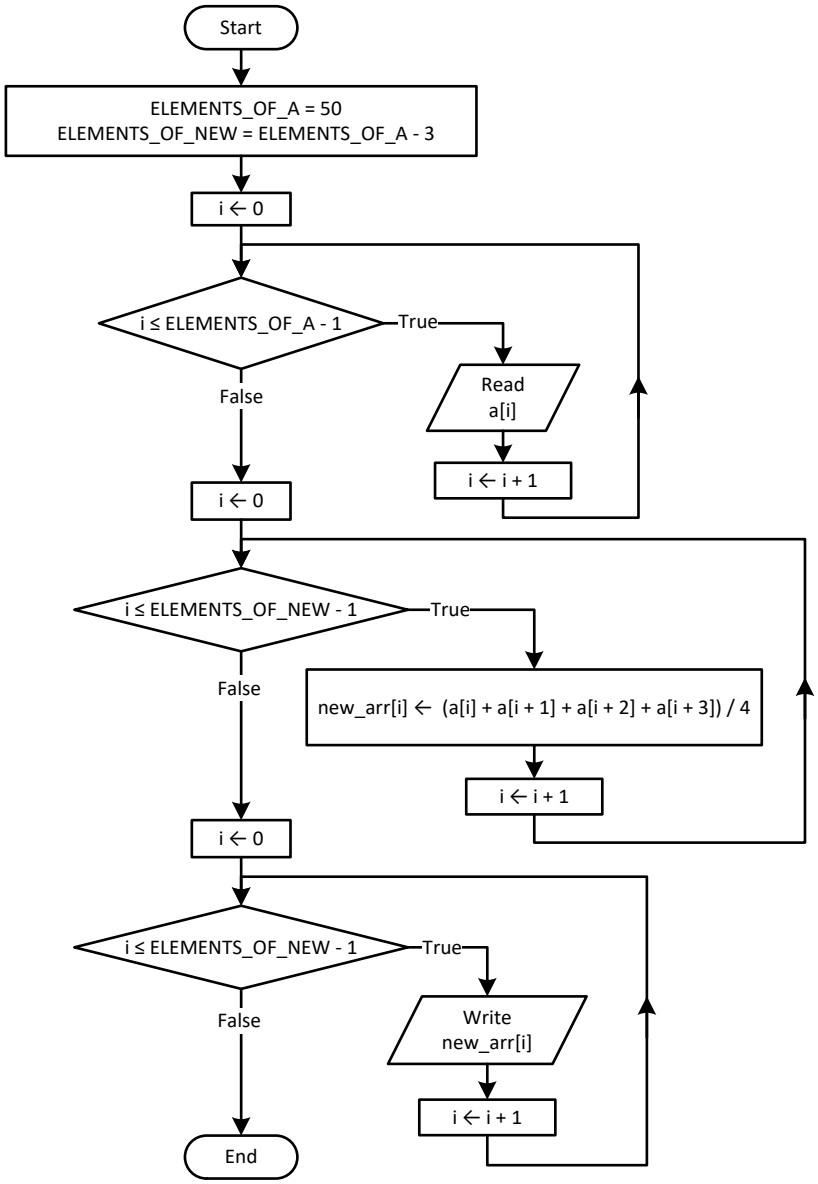
7. Solution

```
max_i = 0;
max_j = 0;
for (i = 0; i <= CITIES - 1; i++) {
  for (j = 0; j <= CITIZENS - 1; j++) {
    if (a[i][j] > a[max_i][max_j]) {
      max_i = i;
      max_j = j;
    }
  }
}

System.out.println(a[max_i][max_j]);
```

8. Solution

```
for (i = 0; i <= ROWS - 1; i++) {
  for (j = 0; j <= COLUMNS - 1; j++) {
    a[i][j] = Double.parseDouble(cin.nextLine());
    while (a[i][j] == 0) {
      System.out.println("Error");
      a[i][j] = Double.parseDouble(cin.nextLine());
    }
  }
}
```

9. Solution

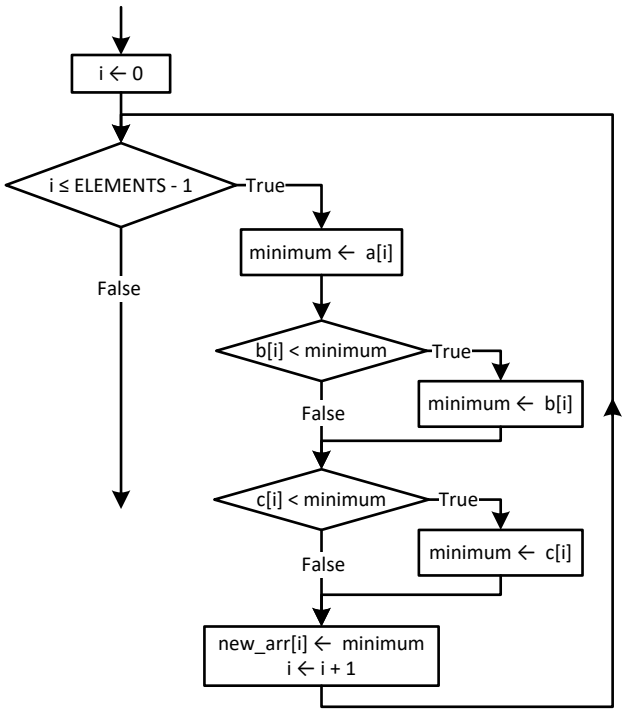



```
static final int ELEMENTS_OF_A = 50;
static final int ELEMENTS_OF_NEW = ELEMENTS_OF_A - 3;

public static void main(String[] args) {
  int i;

  double[] a = new double[ELEMENTS_OF_A];
  for (i = 0; i <= ELEMENTS_OF_A - 1; i++) {
    a[i] = Double.parseDouble(cin.nextLine());
  }

  double[] new_arr = new double[ELEMENTS_OF_NEW];
  for (i = 0; i <= ELEMENTS_OF_NEW - 1; i++) {
    new_arr[i] = (a[i] + a[i + 1] + a[i + 2] + a[i + 3]) / 4;
```

```
  }

  for (i = 0; i <= ELEMENTS_OF_NEW - 1; i++) {
    System.out.println(new_arr[i] + "\t");
  }
}
```

10. Solution

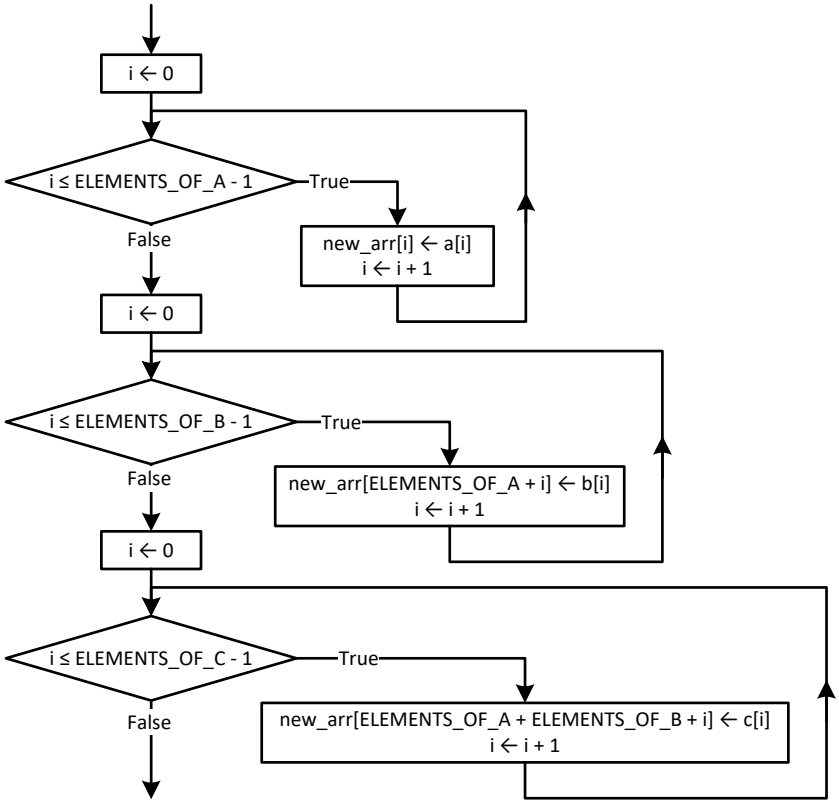
```
static final int ELEMENTS = 15;

public static void main(String[] args) {
  int i;
  double minimum

  double[] a = new double[ELEMENTS];
  for (i = 0; i <= ELEMENTS - 1; i++) {
    a[i] = Double.parseDouble(cin.nextLine());
  }
  double[] b = new double[ELEMENTS];
  for (i = 0; i <= ELEMENTS - 1; i++) {
    b[i] = Double.parseDouble(cin.nextLine());
  }
  double[] c = new double[ELEMENTS];
  for (i = 0; i <= ELEMENTS - 1; i++) {
    c[i] = Double.parseDouble(cin.nextLine());
  }

  double[] new_arr = new double[ELEMENTS];
  for (i = 0; i <= ELEMENTS - 1; i++) {
    minimum = a[i];
    if (b[i] < minimum) {
      minimum = b[i];
    }
    if (c[i] < minimum) {
      minimum = c[i];
    }
    new_arr[i] = minimum;
  }

  for (i = 0; i <= ELEMENTS - 1; i++) {
    System.out.println(new_arr[i]);
  }
}
```



11. Solution

```
static final int ELEMENTS_OF_A = 10;
static final int ELEMENTS_OF_B = 5;
static final int ELEMENTS_OF_C = 15;
static final int ELEMENTS_OF_NEW = ELEMENTS_OF_A + ELEMENTS_OF_B + ELEMENTS_OF_C;

public static void main(String[] args) {
  int i;

  double[] a = new double[ELEMENTS_OF_A];
  for (i = 0; i <= ELEMENTS_OF_A - 1; i++) {
    a[i] = Double.parseDouble(cin.nextLine());
  }
  double[] b = new double[ELEMENTS_OF_B];
  for (i = 0; i <= ELEMENTS_OF_B - 1; i++) {
    b[i] = Double.parseDouble(cin.nextLine());
  }
  double[] c = new double[ELEMENTS_OF_C];
  for (i = 0; i <= ELEMENTS_OF_C - 1; i++) {
    c[i] = Double.parseDouble(cin.nextLine());
  }

  double[] new_arr = new double[ELEMENTS_OF_NEW];
  for (i = 0; i <= ELEMENTS_OF_C - 1; i++) {
    new_arr[i] = c[i];
  }
  for (i = 0; i <= ELEMENTS_OF_B - 1; i++) {
    new_arr[ELEMENTS_OF_C + i] = b[i];
```

```
  }
  for (i = 0; i <= ELEMENTS_OF_A - 1; i++) {
    new_arr[ELEMENTS_OF_B + ELEMENTS_OF_C + i] = a[i];
  }

  //Display array new
  for (i = 0; i <= ELEMENTS_OF_NEW - 1; i++) {
    System.out.print(new_arr[i] + "\t");
  }
}
```



12. Solution

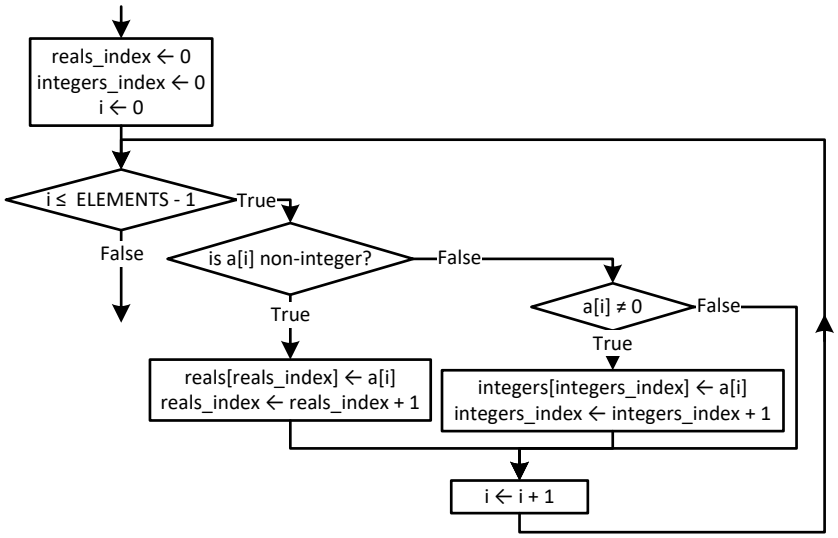
```
static final int COLUMNS_OF_A = 10;
static final int COLUMNS_OF_B = 15;
static final int COLUMNS_OF_C = 20;
static final int ROWS = 5;
static final int COLUMNS = COLUMNS_OF_A + COLUMNS_OF_B + COLUMNS_OF_C;

public static void main(String[] args) {
  int i, j;

  double[][] a = new double[ROWS][COLUMNS_OF_A];
  for (i = 0; i <= ROWS - 1; i++) {
    for (j = 0; j <= COLUMNS_OF_A - 1; j++) {
      a[i][j] = Double.parseDouble(cin.nextLine());
    }
```

```
  }

  double[][] b = new double[ROWS][COLUMNS_OF_B];
  for (i = 0; i <= ROWS - 1; i++) {
    for (j = 0; j <= COLUMNS_OF_B - 1; j++) {
      b[i][j] = Double.parseDouble(cin.nextLine());
    }
  }

  double[][] c = new double[ROWS][COLUMNS_OF_C];
  for (i = 0; i <= ROWS - 1; i++) {
    for (j = 0; j <= COLUMNS_OF_C - 1; j++) {
      c[i][j] = Double.parseDouble(cin.nextLine());
    }
  }

  double[][] new_arr = new double[ROWS][COLUMNS];
  for (i = 0; i <= ROWS - 1; i++) {
    for (j = 0; j <= COLUMNS_OF_A - 1; j++) {
      new_arr[i][j] = a[i][j];
    }
  }
  for (i = 0; i <= ROWS - 1; i++) {
    for (j = 0; j <= COLUMNS_OF_B - 1; j++) {
      new_arr[i][COLUMNS_OF_A + j] = b[i][j];
    }
  }
  for (i = 0; i <= ROWS - 1; i++) {
    for (j = 0; j <= COLUMNS_OF_C - 1; j++) {
      new_arr[i][COLUMNS_OF_A + COLUMNS_OF_B + j] = c[i][j];
    }
  }

  for (i = 0; i <= ROWS - 1; i++) {
    for (j = 0; j <= COLUMNS - 1; j++) {
      System.out.print(new_arr[i][j] + "\t");
    }
    System.out.println();
  }
}
```

13. Solution

```
static final int ELEMENTS = 50;

public static void main(String[] args) {
  int i, integers_index, reals_index;

  double[] a = new double[ELEMENTS];
  for (i = 0; i <= ELEMENTS - 1; i++) {
    a[i] = Double.parseDouble(cin.nextLine());
```

```
  }

  double[] reals = new double[ELEMENTS];
  int[] integers = new int[ELEMENTS];
  reals_index = 0;
  integers_index = 0;
  for (i = 0; i <= ELEMENTS - 1; i++) {
    if (a[i] != (int)(a[i])) {
      reals[reals_index] = a[i];
      reals_index++;
    }
    else if (a[i] != 0) {
      integers[integers_index] = (int)a[i];
      integers_index++;
    }
  }

  for (i = 0; i <= reals_index - 1; i++) {
    System.out.print(reals[i] + "\t");
  }

  System.out.println();
  for (i = 0; i <= integers_index - 1; i++) {
    System.out.print(integers[i] + "\t");
  }
}
```



14. Solution

```
static final int ELEMENTS = 50;

public static void main(String[] args) {
  int digit1, digit2, digit3, i, k, r;
```

```
  int[] a = new int[ELEMENTS];
  for (i = 0; i <= ELEMENTS - 1; i++) {
    a[i] = Integer.parseInt(cin.nextLine());
  }

  int[] b = new int[ELEMENTS];
  k = 0;
  for (i = 0; i <= ELEMENTS - 1; i++) {
    digit3 = a[i] % 10;
    r = (int)(a[i] / 10);
    digit2 = r % 10;
    digit1 = (int)(r / 10);

    if (digit1 < digit2 && digit2 < digit3) {
      b[k] = a[i];
      k++;
    }
  }

  for (i = 0; i <= k - 1; i++) {
    System.out.print(b[i] + "\t");
  }
}
```

15. Solution

```
static final int PRODUCTS = 10;
static final int CITIZENS = 200;

public static void main(String[] args) {
  int count_B, i, j, maximum;

  String[] prod_names = new String[PRODUCTS];
  String[][] answers = new String[PRODUCTS][CITIZENS];
  for (i = 0; i <= PRODUCTS - 1; i++) {
    prod_names[i] = cin.nextLine();
    for (j = 0; j <= CITIZENS - 1; j++) {
      answers[i][j] = cin.nextLine();
      while (answers[i][j].compareTo("A") < 0 || answers[i][j].compareTo("D") > 0) {
        System.out.println("Error! ");
        answers[i][j] = cin.nextLine();
      }
    }
  }

  int[] count_A = new int[PRODUCTS];
  for (i = 0; i <= PRODUCTS - 1; i++) {
    count_A[i] = 0;
    for (j = 0; j <= CITIZENS - 1; j++) {
      if (answers[i][j].equals("A")) {
        count_A[i]++;
```

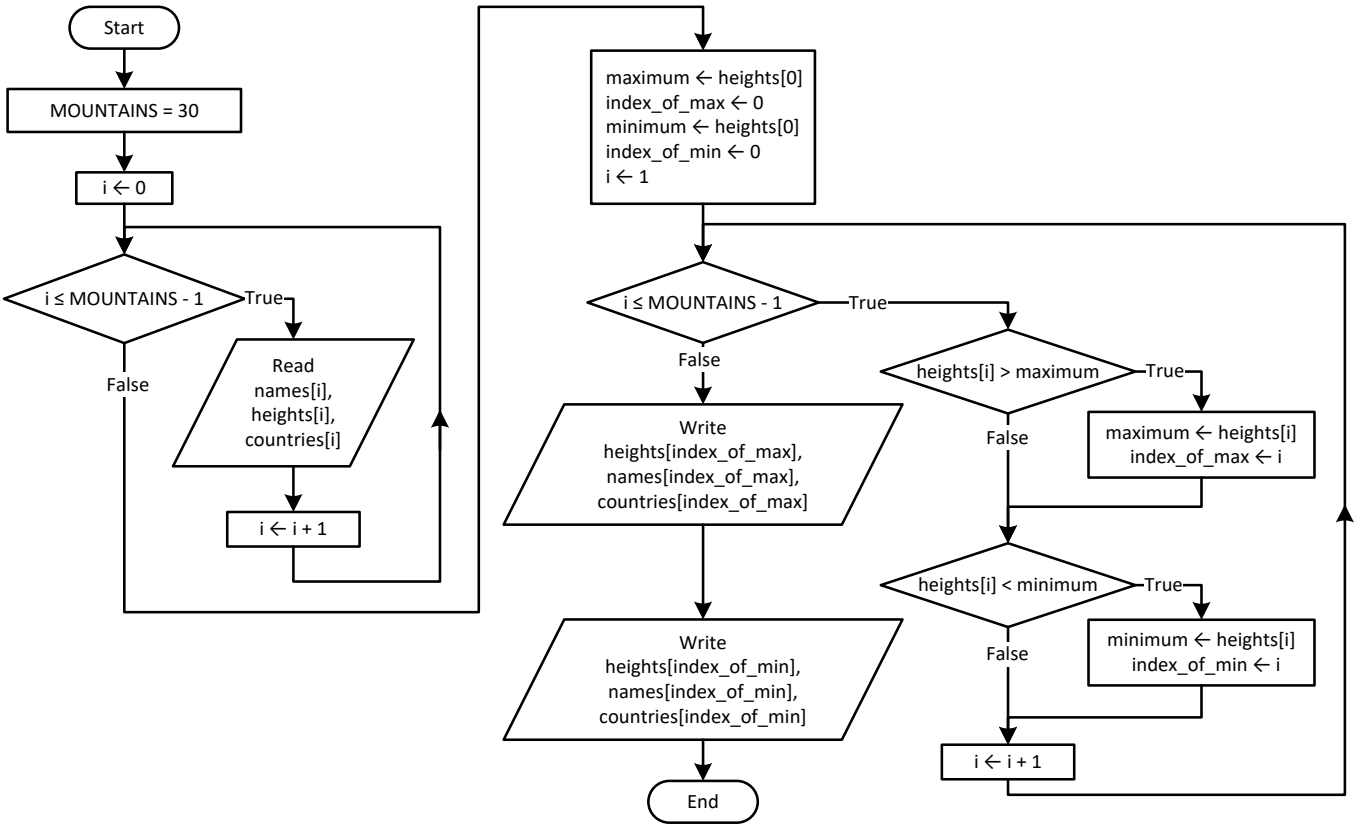
```
      }
    }
    System.out.println(prod_names[i] + ", " + count_A[i]);
  }

  for (j = 0; j <= CITIZENS - 1; j++) {
    count_B = 0;
    for (i = 0; i <= PRODUCTS - 1; i++) {
      if (answers[i][j].equals("B")) {
        count_B++;
      }
    }
    System.out.println(count_B);
  }

  maximum = count_A[0];
  for (i = 1; i <= PRODUCTS - 1; i++) {
    if (count_A[i] > maximum) {
      maximum = count_A[i];
    }
  }
  for (i = 0; i <= PRODUCTS - 1; i++) {
    if (count_A[i] == maximum) {
      System.out.println(prod_names[i]);
    }
  }
}
```

16. Solution

```
static final int US_CITIES = 20;
static final int CANADIAN_CITIES = 20;

public static void main(String[] args) {
  int i, j, min_j;
  double minimum;

  String[] us_names = new String[US_CITIES];
  for (i = 0; i <= US_CITIES - 1; i++) {
    System.out.println("Enter name for US city No " + (i + 1) + ": ");
    us_names[i] = cin.nextLine();
  }

  String[] canadian_names = new String[CANADIAN_CITIES];
  for (j = 0; j <= CANADIAN_CITIES - 1; j++) {
    System.out.println("Enter name for Canadian city No " + (j + 1) + ": ");
    canadian_names[j] = cin.nextLine();
  }

  double[][] distances = new double[US_CITIES][CANADIAN_CITIES];
  for (i = 0; i <= US_CITIES - 1; i++) {
```

```
    for (j = 0; j <= CANADIAN_CITIES - 1; j++) {
      System.out.println("Enter distance between " + us_names[i] + " and " + canadian_names[j] + ": ");
      distances[i][j] = Double.parseDouble(cin.nextLine());
    }
  }

  for (i = 0; i <= US_CITIES - 1; i++) {
    minimum = distances[i][0];
    min_j = 0;
    for (j = 1; j <= CANADIAN_CITIES - 1; j++) {
      if (distances[i][j] < minimum) {
        minimum = distances[i][j];
        min_j = j;
      }
    }
    System.out.println("Closest Canadian city to " + us_names[i] + " is " + canadian_names[min_j]);
  }
}
```

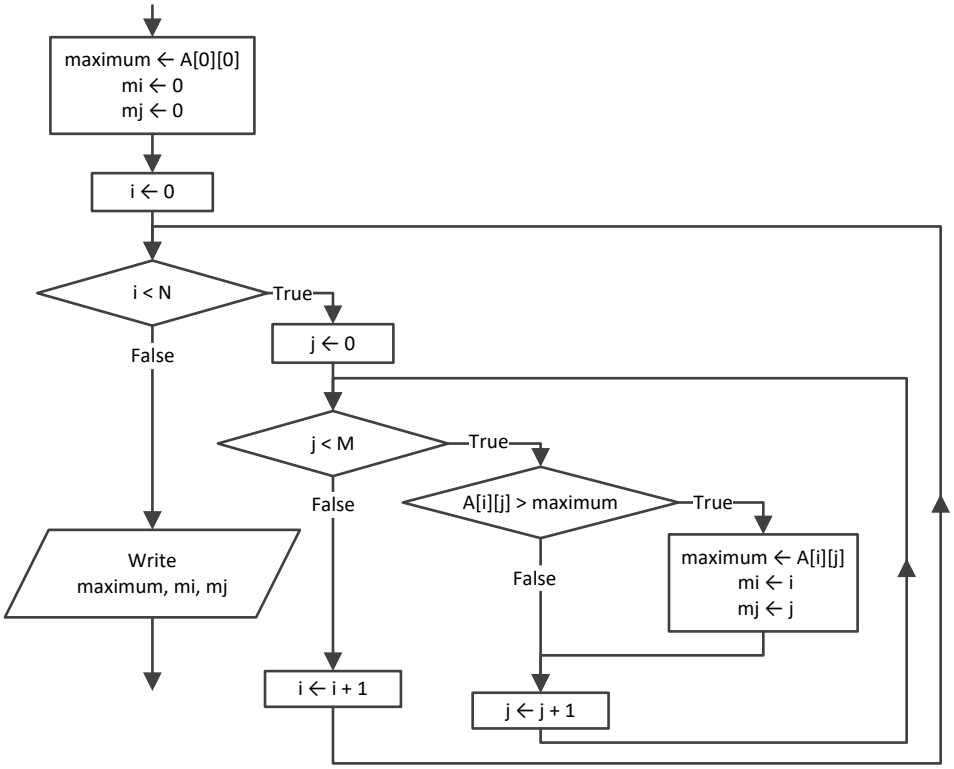
17. Solution




```
static final int MOUNTAINS = 30;

public static void main(String[] args) {
  int i, index_of_max, index_of_min;
  double maximum, minimum;
```

```
  String[] names = new String[MOUNTAINS];
  double[] heights = new double[MOUNTAINS];
  string[] countries = new string[MOUNTAINS];
  for (i = 0; i <= MOUNTAINS - 1; i++) {
    names[i] = cin.nextLine();
    heights[i] = Double.parseDouble(cin.nextLine());
    countries[i] = cin.nextLine();
  }

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

  System.out.println(heights[index_of_max] +", "+ names[index_of_max] +", "+ countries[index_of_max]);
  System.out.println(heights[index_of_min] +", "+ names[index_of_min] +", "+ countries[index_of_min]);
}
```

18. Solution



19. Solution

```
static final int TEAMS = 26;
static final int GAMES = 15;

public static void main(String[] args) {
  int i, j, m_i, maximum;

  String[] names = new String[TEAMS];
  String[][] results = new String[TEAMS][GAMES];
  for (i = 0; i <= TEAMS - 1; i++) {
    names[i] = cin.nextLine();
    for (j = 0; j <= GAMES - 1; j++) {
      results[i][j] = cin.nextLine();
    }
  }

  int[] points = new int[TEAMS];
  for (i = 0; i <= TEAMS - 1; i++) {
    points[i] = 0;
    for (j = 0; j <= GAMES - 1; j++) {
      if (results[i][j].equals("W")) {
        points[i] += 3;
      }
      else if (results[i][j].equals("T")) {
```

```
        points[i] += 1;
      }
    }
  }

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

  System.out.println(names[m_i]);
}
```

20. Solution

```
static final int OBJECTS = 10;
static final int FALLS = 20;

public static void main(String[] args) {
  int i, j;
  double maxi, mini;

  double[][] heights = new double[OBJECTS][FALLS];
  double[][] times = new double[OBJECTS][FALLS];
  for (i = 0; i <= OBJECTS - 1; i++) {
    for (j = 0; j <= FALLS - 1; j++) {
      heights[i][j] = Double.parseDouble(cin.nextLine());
      times[i][j] = Double.parseDouble(cin.nextLine());
    }
  }

  double[][] g = new double[OBJECTS][FALLS];
  for (i = 0; i <= OBJECTS - 1; i++) {
    for (j = 0; j <= FALLS - 1; j++) {
      g[i][j] = 2 * heights[i][j] / Math.pow(times[i][j], 2);
    }
  }

  double[] minimum = new double[OBJECTS];
  double[] maximum = new double[OBJECTS];
  for (i = 0; i <= OBJECTS - 1; i++) {
    minimum[i] = g[i][0];
    maximum[i] = g[i][0];
    for (j = 1; j <= FALLS - 1; j++) {
      if (g[i][j] < minimum[i]) {
        minimum[i] = g[i][j];
      }
```

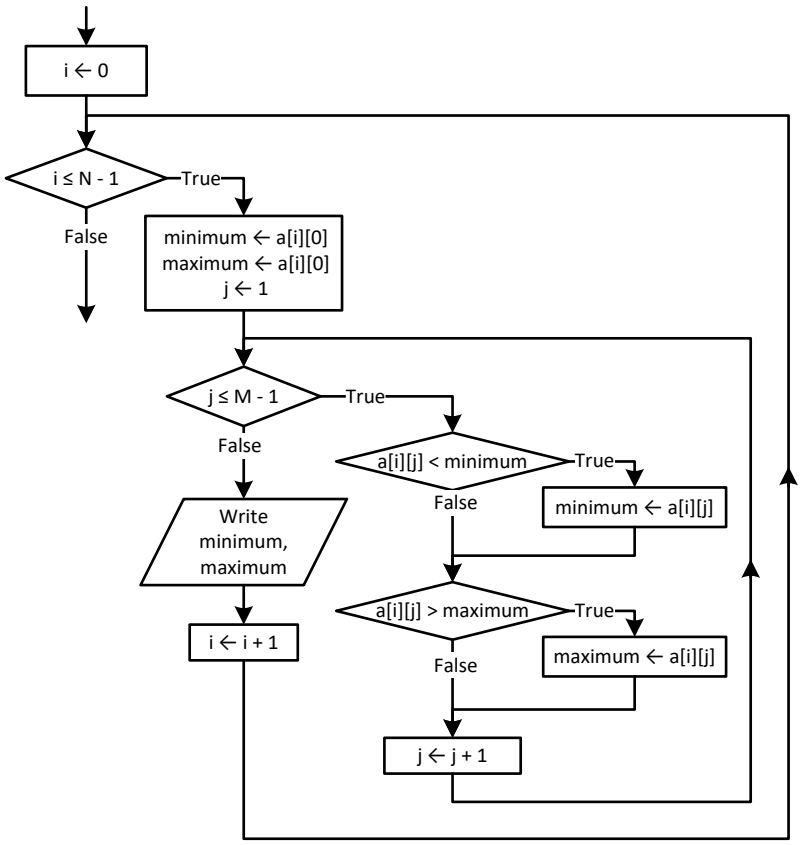
```
      if (g[i][j] > maximum[i]) {
        maximum[i] = g[i][j];
      }
    }
  }

  for (i = 0; i <= OBJECTS - 1; i++) {
    System.out.println(minimum[i] + ", " + maximum[i]);
  }

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

  System.out.println(mini + ", " + maxi);
}
```

21. Solution

```
static final int STATIONS = 10;
static final int DAYS = 365;

public static void main(String[] args) {
  int i, j, m_i;
  double minimum;

  String[] names = new String[STATIONS];
  double[][] co2 = new double[STATIONS][DAYS];
  for (i = 0; i <= STATIONS - 1; i++) {
    names[i] = cin.nextLine();
    for (j = 0; j <= DAYS - 1; j++) {
      co2[i][j] = Double.parseDouble(cin.nextLine());
    }
  }

  double[] average = new double[STATIONS];
  for (i = 0; i <= STATIONS - 1; i++) {
    average[i] = 0;
    for (j = 0; j <= DAYS - 1; j++) {
      average[i] += co2[i][j];
    }
    average[i] /= DAYS;
  }
```

```
  minimum = average[0];
  m_i = 0;
  for (i = 1; i <= STATIONS - 1; i++) {
    if (average[i] < minimum) {
      minimum = average[i];
      m_i = i;
    }
  }
  System.out.println(names[m_i]);
}
```

22. Solution



23. Solution

First approach

```
static final int ROWS = 20;
static final int COLUMNS = 30;

public static void main(String[] args) {
  int i, j;

  double[][] b = new double[ROWS][COLUMNS];
  for (i = 0; i <= ROWS - 1; i++) {
    for (j = 0; j <= COLUMNS - 1; j++) {
```

```
      b[i][j] = Double.parseDouble(cin.nextLine());
    }
  }

  double[] minimum = new double[COLUMNS];
  double[] maximum = new double[COLUMNS];
  for (j = 0; j <= COLUMNS - 1; j++) {
    minimum[j] = b[0][j];
    maximum[j] = b[0][j];
    for (i = 1; i <= ROWS - 1; i++) {
      if (b[i][j] < minimum[j]) {
        minimum[j] = b[i][j];
      }
      if (b[i][j] > maximum[j]) {
        maximum[j] = b[i][j];
      }
    }
  }

  for (j = 0; j <= COLUMNS - 1; j++) {
    System.out.println(minimum[j] + " " + maximum[j]);
  }
}
```

Second approach

```
static final int ROWS = 20;
static final int COLUMNS = 30;

public static void main(String[] args) {
  int i, j;
  double minimum, maximum;

  double[][] b = new double[ROWS][COLUMNS];
  for (i = 0; i <= ROWS - 1; i++) {
    for (j = 0; j <= COLUMNS - 1; j++) {
      b[i][j] = Double.parseDouble(cin.nextLine());
    }
  }

  for (j = 0; j <= COLUMNS - 1; j++) {
    minimum = b[0][j];
    maximum = b[0][j];
    for (i = 1; i <= ROWS - 1; i++) {
      if (b[i][j] < minimum) {
        minimum = b[i][j];
      }
      if (b[i][j] > maximum) {
        maximum = b[i][j];
      }
    }
    System.out.println(minimum + " " + maximum);
  }
```

```
}
```

24. Solution

```
static final int TEAMS = 20;
static final int GAMES = 10;

public static void main(String[] args) {
  int i, j, m, n, temp;
  boolean swaps;
  String temp_str;

  String[] names = new String[TEAMS];
  String[][] results = new String[TEAMS][GAMES];
  for (i = 0; i <= TEAMS - 1; i++) {
    System.out.print("Enter team name: ");
    names[i] = cin.nextLine();
    for (j = 0; j <= GAMES - 1; j++) {
      System.out.println("Enter result for team " + names[i] + " for game No " + (j + 1) + ": ");
      results[i][j] = cin.nextLine();
      while (!results[i][j].equals("W") && !results[i][j].equals("L") && !results[i][j].equals("T")) {
        System.out.print("Error! Enter only value W, L, or T: ");
        results[i][j] = cin.nextLine();
      }
    }
  }

  int[] points = new int[TEAMS];
  for (i = 0; i <= TEAMS - 1; i++) {
    points[i] = 0;
    for (j = 0; j <= GAMES - 1; j++) {
      if (results[i][j].equals("W")) {
        points[i] += 3;
      }
      else if (results[i][j].equals("T")) {
        points[i] += 1;
      }
    }
  }

  for (m = 1; m <= TEAMS - 1; m++) {
    swaps = false;
    for (n = TEAMS - 1; n >= m; n--) {
      if (points[n] > points[n - 1]) {
        temp = points[n];
        points[n] = points[n - 1];
        points[n - 1] = temp;

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

```
        swaps = true;
      }
    }
    if (!swaps) break;
  }

  System.out.println("Gold: " + names[0]);
  System.out.println("Silver: " + names[1]);
  System.out.println("Bronze: " + names[2]);
}
```

25. Solution

```
static final int PEOPLE = 50;

public static void main(String[] args) {
  int i, m, n;
  double temp;
  String temp_str;

  String[] names = new String[PEOPLE];
  double[] heights = new double[PEOPLE];
  for (i = 0; i <= PEOPLE - 1; i++) {
    System.out.print("Enter name for person No. " + (i + 1) + ": ");
    names[i] = cin.nextLine();
    System.out.print("Enter height for person No. " + (i + 1) + ": ");
    heights[i] = Double.parseDouble(cin.nextLine());
  }

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

        temp_str = names[n];
        names[n] = names[n - 1];
        names[n - 1] = temp_str;
      }
      else if (heights[n] == heights[n - 1]) {
        if (names[n].compareTo(names[n - 1]) < 0) {
          temp_str = names[n];
          names[n] = names[n - 1];
          names[n - 1] = temp_str;
        }
      }
    }
  }
```

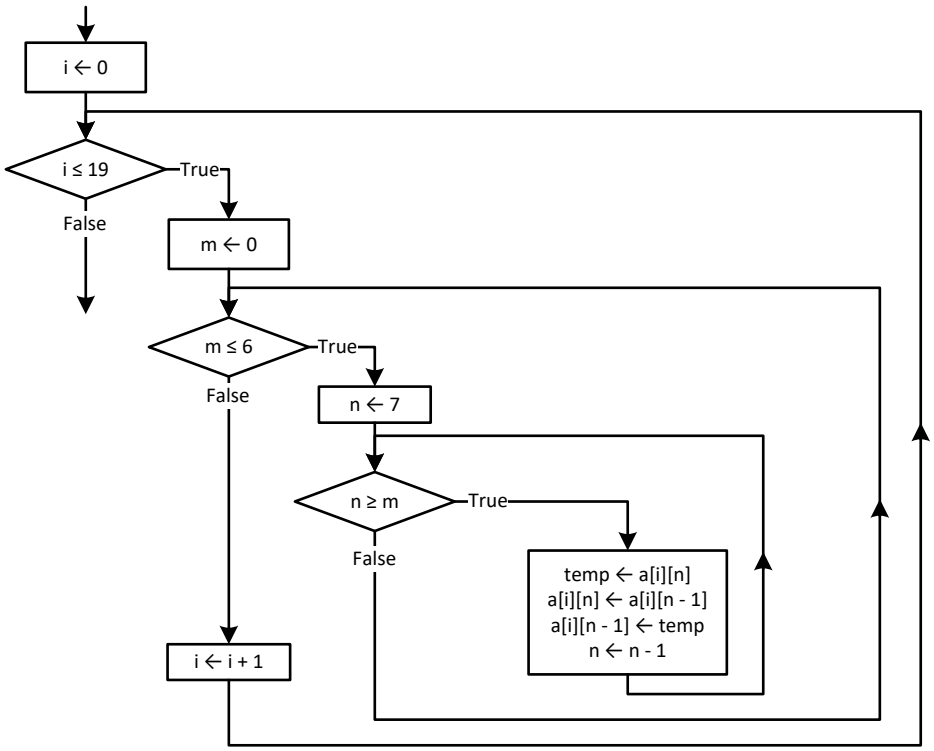
```
  for (i = 0; i <= PEOPLE - 1; i++) {
    System.out.println(heights[i] + "\t" + names[i]);
  }
}
```

26. Solution

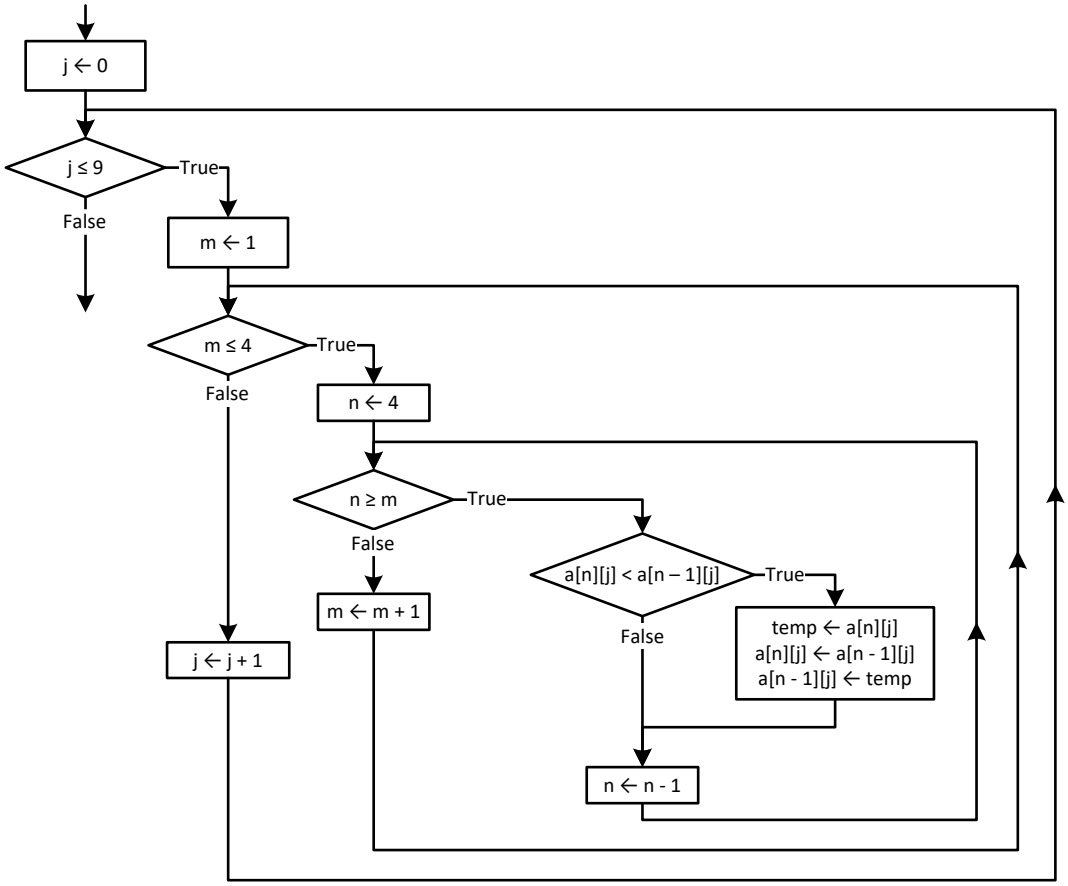
```
static final int ARTISTS = 12;
static final int JUDGES = 10;

public static void main(String[] args) {
  int i, j, m, maximum, minimum, n, temp;
  String temp_str;

  String[] artist_names = new String[ARTISTS];
  int[][] score = new int[ARTISTS][JUDGES];
  for (i = 0; i <= ARTISTS - 1; i++) {
    System.out.println("Enter name for artist No " + (i + 1) + ": ");
    artist_names[i] = cin.nextLine();
    for (j = 0; j <= JUDGES - 1; j++) {
      System.out.print("Enter score for artist: " + artist_names[i]);
      System.out.println(" gotten from judge No " + (j + 1) + ": ");
      score[i][j] = Integer.parseInt(cin.nextLine());
    }
  }

  int[] total = new int[ARTISTS];
  for (i = 0; i <= ARTISTS - 1; i++) {
    total[i] = 0;
    for (j = 1; j <= JUDGES - 1; j++) {
      total[i] += score[i][j];
    }
  }

  for (i = 0; i <= ARTISTS - 1; i++) {
    minimum = score[i][0];
    maximum = score[i][0];
    for (j = 1; j <= JUDGES - 1; j++) {
      if (score[i][j] < minimum) {
        minimum = score[i][j];
      }
      if (score[i][j] > maximum) {
        maximum = score[i][j];
      }
    }
    total[i] = total[i] - minimum - maximum;
    System.out.println(total[i]);
  }

  for (m = 1; m <= ARTISTS - 1; m++) {
    for (n = ARTISTS - 1; n >= m; n--) {
```

```
      if (total[n] > total[n - 1]) {
        temp = total[n];
        total[n] = total[n - 1];
        total[n - 1] = temp;

        temp_str = artist_names[n];
        artist_names[n] = artist_names[n - 1];
        artist_names[n - 1] = temp_str;
      }
      else if (total[n] == total[n - 1]) {
        if (artist_names[n].compareTo(artist_names[n - 1]) < 0) {
          temp_str = artist_names[n];
          artist_names[n] = artist_names[n - 1];
          artist_names[n - 1] = temp_str;
        }
      }
    }
  }

  for (i = 0; i <= ARTISTS - 1; i++) {
    System.out.println(artist_names[i] + ", " + total[i]);
  }
}
```

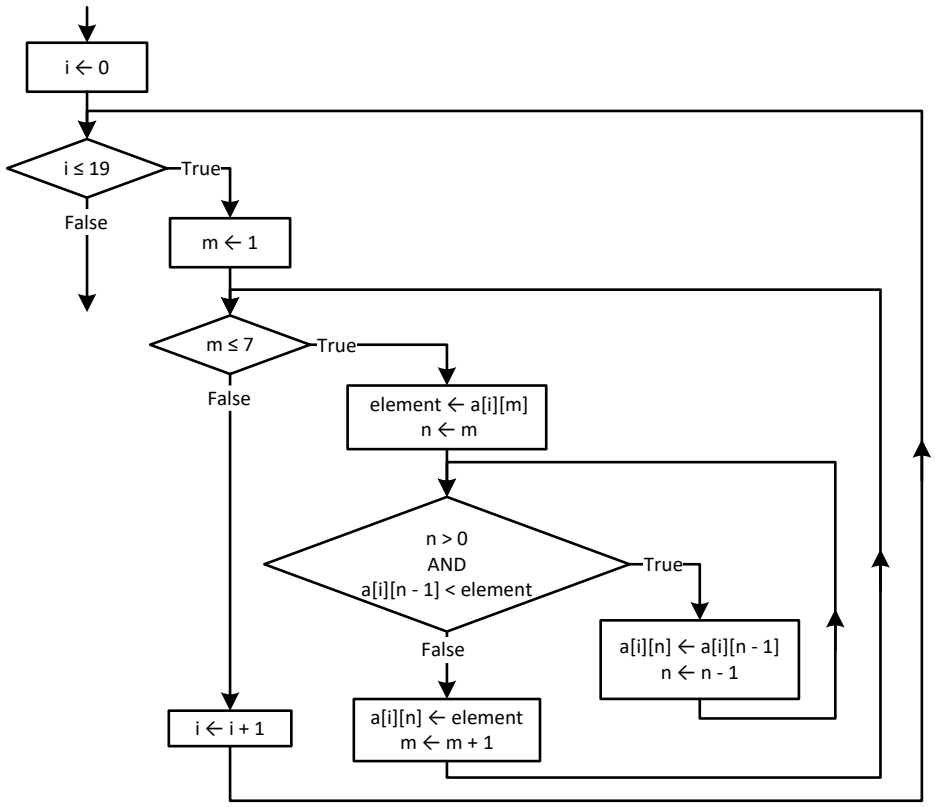
27. Solution



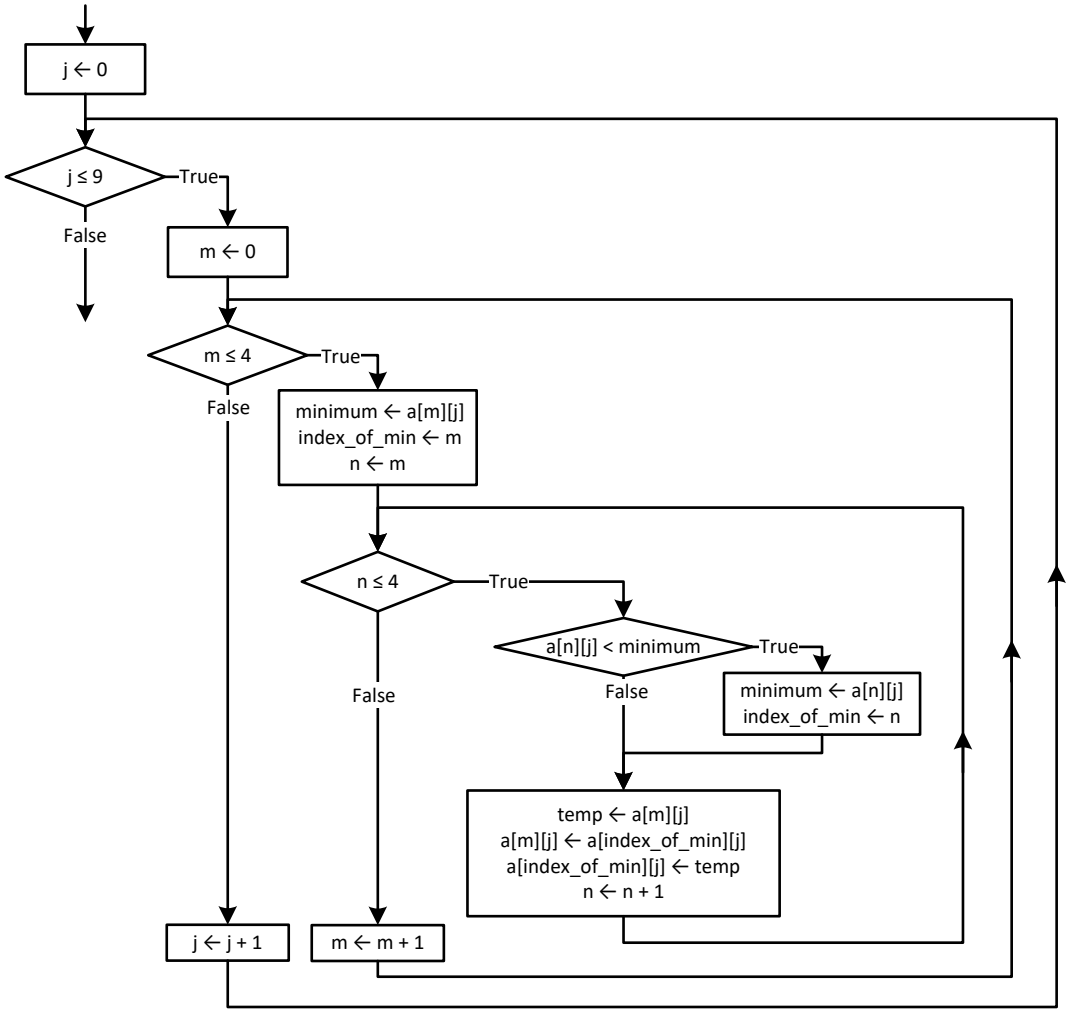
28. Solution


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
temp ← a[n][j]
a[n][j] ← a[n - 1][j]
a[n - 1][j] ← temp
j ← j + 1
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
AND
a[i][n - 1] < element
True
False
a[i][n] ← a[i][n - 1]
n ← n - 1
a[i][n] ← element
m ← m + 1
i ← i + 1

30. Solution



31. Solution

```
static final int PEOPLE = 10;
static final int PUZZLES = 8;

public static void main(String[] args) {
  int i, index_of_min, j, m, n;
  double minimum, temp;
  String temp_str;

  String[] names = new String[PEOPLE];
  double[][] times = new double[PEOPLE][PUZZLES];
  for (i = 0; i <= PEOPLE - 1; i++) {
    names[i] = cin.nextLine();
    for (j = 0; j <= PUZZLES - 1; j++) {
      times[i][j] = Double.parseDouble(cin.nextLine());
    }
  }
```

```
for (i = 0; i <= PEOPLE - 1; i++) {
  for (m = 0; m <= PUZZLES - 1; m++) {
    minimum = times[i][m];
    index_of_min = m;
    for (n = m; n <= PUZZLES - 1; n++) {
      if (times[i][n] < minimum) {
        minimum = times[i][n];
        index_of_min = n;
      }
    }
    temp = times[i][m];
    times[i][m] = times[i][index_of_min];
    times[i][index_of_min] = temp;
  }
}

for (i = 0; i <= PEOPLE - 1; i++) {
  System.out.println(names[i]);
  for (j = 0; j <= 2; j++) {
    System.out.println(times[i][j]);
  }
}

double[] average = new double[PEOPLE];
for (i = 0; i <= PEOPLE - 1; i++) {
  average[i] = 0;
  for (j = 0; j <= PUZZLES - 1; j++) {
    average[i] += times[i][j];
  }
  average[i] /= PUZZLES;
}

for (m = 0; m <= PEOPLE - 1; m++) {
  minimum = average[m];
  index_of_min = m;
  for (n = m; n <= PEOPLE - 1; n++) {
    if (average[n] < minimum) {
      minimum = average[n];
      index_of_min = n;
    }
  }
  temp = average[m];
  average[m] = average[index_of_min];
  average[index_of_min] = temp;

  temp_str = names[m];
  names[m] = names[index_of_min];
  names[index_of_min] = temp_str;
}
```

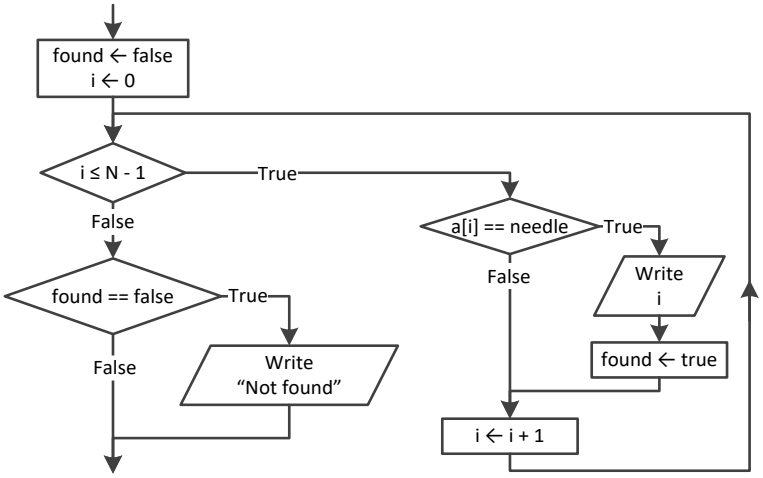
```
  System.out.println(names[0] + ", " + names[1] + ", " + names[2]);
}
```

32. Solution

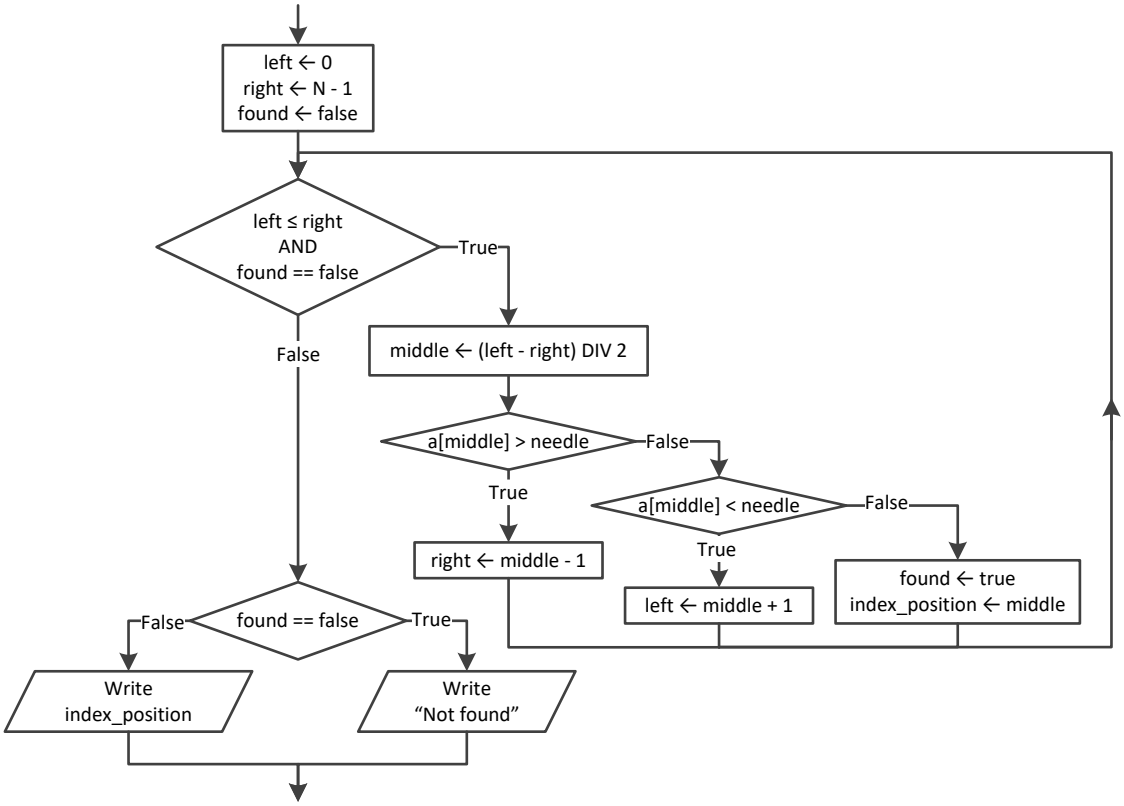
```
static final int AREAS = 5;
static final int HOURS = 48;

public static void main(String[] args) {
  int i, j, m, m_i, m_j, n;
  double maximum, element_1;
  String element_2;

  String[] names = new String[AREAS];
  double[][] CO2 = new double[AREAS][HOURS];
  for (i = 0; i <= AREAS - 1; i++) {
    names[i] = cin.nextLine();
    for (j = 0; j <= HOURS - 1; j++) {
      CO2[i][j] = Double.parseDouble(cin.nextLine());
    }
  }

  double[] average_per_hour = new double[AREAS];
  for (i = 0; i <= AREAS - 1; i++) {
    average_per_hour[i] = 0;
    for (j = 0; j <= HOURS - 1; j++) {
      average_per_hour[i] += CO2[i][j];
    }
    average_per_hour[i] /= HOURS;
  }

  for (i = 0; i <= AREAS - 1; i++) {
    System.out.println(names[i] + ", " + average_per_hour[i]);
  }

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

  for (j = 0; j <= HOURS - 1; j++) {
    System.out.println(average_per_city[j]);
  }

  maximum = average_per_city[0];
  m_j = 0;
  for (j = 1; j <= HOURS - 1; j++) {
```

```
    if (average_per_city[j] > maximum) {
      maximum = average_per_city[j];
      m_j = j;
    }
  }
  System.out.println(m_j);

  maximum = CO2[0][0];
  m_i = 0;
  m_j = 0;
  for (i = 0; i <= AREAS - 1; i++) {
    for (j = 0; j <= HOURS - 1; j++) {
      if (CO2[i][j] > maximum) {
        maximum = CO2[i][j];
        m_i = i;
        m_j = j;
      }
    }
  }
  System.out.println(m_j + ", " + names[m_i]);

  for (m = 1; m <= AREAS - 1; m++) {
    element_1 = average_per_hour[m];
    element_2 = names[m];

    n = m;
    while (n > 0 && average_per_hour[n - 1] < element_1) {
      average_per_hour[n] = average_per_hour[n - 1];
      names[n] = names[n - 1];
      n--;
    }

    average_per_hour[n] = element_1;
    names[n] = element_2;
  }

  System.out.println(names[0] + ", " + names[1]  + ", " +  names[2]);
}
```

33. Solution



34. Solution



35. Solution

```
static final int TEAMS = 10;
static final int GAMES = 16;

public static void main(String[] args) {
```

```
  int i, j, total;
  String needle;

  String[] names = new String[TEAMS];
  int[][] goals_scored = new int[TEAMS][GAMES];
  int[][] goals_let_in = new int[TEAMS][GAMES];
  for (i = 0; i <= TEAMS - 1; i++) {
    System.out.print("Enter team name: ");
    names[i] = cin.nextLine();
    for (j = 0; j <= GAMES - 1; j++) {
      System.out.print("Enter goals scored: ");
      goals_scored[i][j] = Integer.parseInt(cin.nextLine());
      while (goals_scored[i][j] < 0) {
        System.out.print("Error! Enter goals scored: ");
        goals_scored[i][j] = Integer.parseInt(cin.nextLine());
      }

      System.out.print("Enter goals let in: ");
      goals_let_in[i][j] = Integer.parseInt(cin.nextLine());
      while (goals_let_in[i][j] < 0) {
        System.out.print("Error! Enter goals let in: ");
        goals_let_in[i][j] = Integer.parseInt(cin.nextLine());
      }
    }
  }

  System.out.print("Enter a team to search: ");
  needle = cin.nextLine();

  i = 0;
  while (i < TEAMS - 1 && !names[i].equals(needle)) {
    i++;
  }

  if (!names[i].equals(needle)) {
    System.out.println("This team does not exist");
  }
  else {
    total = 0;
    for (j = 0; j <= GAMES - 1; j++) {
      if (goals_scored[i][j] > goals_let_in[i][j]) {
        total += 3;
      }
      else if (goals_scored[i][j] == goals_let_in[i][j]) {
        total += 1;
      }
    }
    System.out.println(total);
  }
}
```

36. Solution

```
static final int CLASS1 = 20;
static final int CLASS2 = 25;

public static void main(String[] args) {
  int i, left, m, middle, n, right;
  String temp, needle;
  boolean found;

  System.out.println("Class 1");
  String[] names1 = new String[CLASS1];
  for (i = 0; i <= CLASS1 - 1; i++) {
    System.out.print("Enter name: ");
    names1[i] = cin.nextLine();
  }
  System.out.println("Class 2");
  String[] names2 = new String[CLASS2];
  for (i = 0; i <= CLASS2 - 1; i++) {
    System.out.print("Enter name: ");
    names2[i] = cin.nextLine();
  }

  //Bubble sort
  for (m = 1; m <= CLASS1 - 1; m++) {
    for (n = CLASS1 - 1; n >= m; n--) {
      if (names1[n].compareTo(names1[n - 1]) < 0) {
        temp = names1[n];
        names1[n] = names1[n - 1];
        names1[n] = temp;
      }
    }
  }
  for (m = 1; m <= CLASS2 - 1; m++) {
    for (n = CLASS2 - 1; n >= m; n--) {
      if (names2[n].compareTo(names2[n - 1]) < 0) {
        temp = names2[n];
        names2[n] = names2[n - 1];
        names2[n] = temp;
      }
    }
  }

  System.out.println("\nClass 1");
  for (i = 0; i <= CLASS1 - 1; i++) {
    System.out.println(names1[i]);
  }
  System.out.println("\nClass 2");
  for (i = 0; i <= CLASS2 - 1; i++) {
    System.out.println(names2[i]);
  }
```

```
  System.out.print("Enter a name to search: ");
  needle = cin.nextLine();

  left = 0;
  right = CLASS1 - 1;
  found = false;
  while (left <= right && !found) {
    middle = (int)((left + right) / 2);

    if (names1[middle].compareTo(needle) > 0) {
      right = middle - 1;
    }
    else if (names1[middle].compareTo(needle) < 0) {
      left = middle + 1;
    }
    else {
      found = true;
    }
  }

  if (found) {
    System.out.println("Student found in Class No 1");
  }
  else {
    left = 0;
    right = CLASS2 - 1;
    while (left <= right && !found) {
      middle = (int)((left + right) / 2);

      if (names2[middle].compareTo(needle) > 0) {
        right = middle - 1;
      }
      else if (names2[middle].compareTo(needle) < 0) {
        left = middle + 1;
      }
      else {
        found = true;
      }
    }

    if (found) {
      System.out.println("Student found in Class No 2");
    }
    else {
      System.out.println("Student not found in either class");
    }
  }
}
```

37. Solution

```
System.out.print("Enter username: ");
usr = cin.nextLine().toUpperCase();
System.out.print("Enter password: ");
pwd = cin.nextLine().toUpperCase();

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

if (usernames[i].toUpperCase().equals(usr) && passwords[i].toUpperCase().equals(pwd)) {
  System.out.println("Login OK!");
}
else {
  System.out.println("Login Failed!");
}
```

38. Solution

```
System.out.print("Enter a value to search: ");
value_str = cin.nextLine();

found = false;

for (i = 0; i <= 999; i++) {
  if (names[i].equals(value_str)) {
    System.out.println(SSNs[i]);
    found = true;
  }
}


if (!found) {
  value = Integer.parseInt(value_str);
  i = 0;
  while (i < 999 && !SSNs[i].equals(value)) {
    i++;
  }

  if (SSNs[i].equals(value)) {
    found = true;
    System.out.println(names[i]);
  }
}

if (!found) {
  System.out.println("This value does not exist");
}
```

39. Solution

```
static final int STUDENTS = 12;
static final int LESSONS = 6;

public static void main(String[] args) {
  int i, j;
  boolean found, failure;

  int[][] grades = new int[STUDENTS][LESSONS];
  for (i = 0; i <= STUDENTS - 1; i++) {
    for (j = 0; j <= LESSONS - 1; j++) {
      do {
        grades[i][j] = Integer.parseInt(cin.nextLine());
        failure = false;
        if (grades[i][j] < 0) {
          System.out.println("Error! You entered a negative value");
          failure = true;
        }
        else if (grades[i][j] > 100) {
          System.out.println("Error! You entered a value grater than 100");
          failure = true;
        }
      } while (failure);
    }
  }

  double[] average = new double[STUDENTS];
  for (i = 0; i <= STUDENTS - 1; i++) {
    average[i] = 0;
    for (j = 0; j <= LESSONS - 1; j++) {
      average[i] += grades[i][j];
    }
    average[i] /= LESSONS;
  }

  found = false;
  for (i = 0; i <= STUDENTS - 1; i++) {
    if (average[i] < 70) {
      found = true;
      break;
    }
  }

  if (found) {
    System.out.println("There is at least one student that has an average value below 70");
  }
}
```

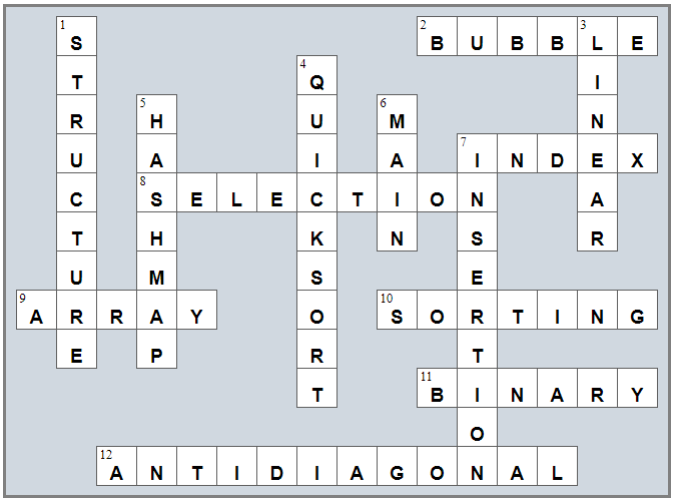
40. Solution

```
public static void main(String[] args) {
  HashMap<String, String> morseAlphabet = new HashMap<>();
  String word, letter;
  int i;

  morseAlphabet.put("A", ".-");
  morseAlphabet.put("B", "-...");
  morseAlphabet.put("C", "-.-.");
  morseAlphabet.put("D", "-..");
  morseAlphabet.put("E", ".");
  morseAlphabet.put("F", "..-.");
  morseAlphabet.put("G", "--.");
  morseAlphabet.put("H", "....");
  morseAlphabet.put("I", "..");
  morseAlphabet.put("J", ".---");
  morseAlphabet.put("K", "-.-");
  morseAlphabet.put("L", ".-..");
  morseAlphabet.put("M", "--");
  morseAlphabet.put("N", "-.");
  morseAlphabet.put("O", "---");
  morseAlphabet.put("P", ".--.");
  morseAlphabet.put("Q", "--.-");
  morseAlphabet.put("R", ".-.");
  morseAlphabet.put("S", "...");
  morseAlphabet.put("T", "-");
  morseAlphabet.put("U", "..-");
  morseAlphabet.put("V", "...-");
  morseAlphabet.put("W", ".--");
  morseAlphabet.put("X", "-..-");
  morseAlphabet.put("Y", "-.--");
  morseAlphabet.put("Z", "--..");
  morseAlphabet.put(" ", "/");

  System.out.print("Enter a word: ");
  word = cin.nextLine();

  for (i = 0; i <= word.length() - 1; i++) {
    letter = "" + word.charAt(i);
    System.out.print(morseAlphabet.get(letter.toUpperCase()) + " ");
  }
}
```

Review in “Data Structures in Java”

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. true
18. true
19. false
20. true
21. true
22. true
23. true
24. false
25. true
26. false
27. true
28. false
29. true
30. true
31. true
32. true
33. false

36.9 Review Exercises

1. Solution

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

2. Solution

Step	Statement	Main Code		Method sum_digits()		
		s	i	a	d1	d2
1	s = 0	**0**	?			
2	i = 25	0	**25**			
3	i <= 27	true				
4	s += sum_digits(i)			**25**	?	?
5	d1 = a % 10			25	**5**	?

6	`d2 = (int)(a / 10)`			25	5	**2**
7	`return d1 + d2`	**7**	25			
8	`i++`	7	**26**			
9	`i <= 27`	true				
10	`s += sum_digits(i)`			**26**	?	?
11	`d1 = a % 10`			26	**6**	?
12	`d2 = (int)(a / 10)`			26	6	**2**
13	`return d1 + d2`	**15**	26			
14	`i++`	15	**27**			
15	`i <= 27`	true				
16	`s += sum_digits(i)`			**27**	?	?
17	`d1 = a % 10`			27	**7**	?
18	`d2 = (int)(a / 10)`			27	7	**2**
19	`return d1 + d2`	**24**	27			
20	`i++`	24	**28**			
21	`i <= 27`	false				
22	`System.out.println(s)`	It displays: 24				

3. Solution

Step	Statement	Main Code		Method sss ()		
		s	i	a	total	k
1	`i = 1`	?	**1**			
2	`s = 0`	**0**	1			
3	`while(i < 6)`	true				
4	`if (i % 2 == 1)`	true				
5	`s += 1`	**1**	1			
6	`i++`	1	**2**			
7	`while(i < 6)`	true				
8	`if (i % 2 == 1)`	false				
9	`s += sss(i)`			**2**	?	?
10	`total = 0`			2	**0**	?
11	`k = 1`			2	0	**1**
12	`k <= a`			true		
13	`total += k`			2	**1**	1
14	`k++`			2	1	**2**
15	`k <= a`			true		
16	`total += k`			2	**3**	2

17	`k++`			2	3	**3**
18	`k <= a`				false	
19	`return total`	**4**	2			
20	`i++`	4	**3**			
21	`while(i < 6)`	true				
22	`if (i % 2 == 1)`	true				
23	`s += 1`	**5**	3			
24	`i++`	5	**4**			
25	`while(i < 6)`	true				
26	`if (i % 2 == 1)`	false				
27	`s += sss(i)`			**4**	?	?
28	`total = 0`			4	**0**	?
29	`k = 1`			4	0	**1**
30	`k <= a`				true	
31	`total += k`			4	**1**	1
32	`k++`			4	1	**2**
33	`k <= a`				true	
34	`total += k`			4	**3**	2
35	`k++`			4	3	**3**
36	`k <= a`				true	
37	`total += k`			4	**6**	4
38	`k++`			4	6	**4**
39	`k <= a`				true	
40	`total += k`			4	**10**	4
41	`k++`			4	10	**5**
42	`k <= a`				false	
43	`return total`	**15**	4			
44	`i++`	15	**5**			
45	`while(i < 6)`	true				
46	`if (i % 2 == 1)`	true				
47	`s += 1`	**16**	5			
48	`i++`	16	**6**			
49	`while(i < 6)`	false				
50	`System.out.println(s)`	It displays: 16				

4. Solution

Step	Statement	Main Code				Method custom_div()	
		k	m	a	x	b	d
1	`k = Integer.parseInt(cin.nextLine())`	**12**	?	?	?		
2	`m = 2`	12	**2**	?	?		
3	`a = 1`	12	2	**1**	?		
4	`while (a < 6)`	true					
5	`if (k % m != 0)`	false					
6	`x = a + m + custom_div(m, a)`					2	1
7	`return (int)((b + d) / 2)`	12	2	1	**4**		
8	`System.out.println(m + " " + a + " " + x)`	It displays: 2 1 4					
9	`a += 2`	12	2	**3**	4		
10	`m++`	12	**3**	3	4		
11	`while (a < 6)`	true					
12	`if (k % m != 0)`	false					
13	`x = a + m + custom_div(m, a)`					3	3
14	`return (int)((b + d) / 2)`	12	3	3	**9**		
15	`System.out.println(m + " " + a + " " + x)`	It displays: 3 3 9					
16	`a += 2`	12	3	**5**	9		
17	`m++`	12	**4**	5	9		
18	`while (a < 6)`	true					
19	`if (k % m != 0)`	false					
20	`x = a + m + custom_div(m, a)`					4	5
21	`return (int)((b + d) / 2)`	12	4	5	**13**		
22	`System.out.println(m + " " + a + " " + x)`	It displays: 4 5 13					
23	`a += 2`	12	4	**7**	13		
24	`m++`	12	**5**	7	13		
25	`while (a < 6)`	false					

5. Solution

Step	Statement	Main Code		void Method display()
		i	x	a
1	`i = 1`	**1**	?	
2	`i <= 5`	true		
3	`x = Integer.parseInt(cin.nextLine())`	1	**3**	

4	`display(x)`			**3**
5	`if (a % 2 == 0)`			false
6	`System.out.println(a + " is odd")`	It displays: 3 is odd		
7	`i++`	**2**	3	
8	`i <= 5`	true		
9	`x = Integer.parseInt(cin.nextLine())`	2	**7**	
10	`display(x)`			**7**
11	`if (a % 2 == 0)`			false
12	`System.out.println(a + " is odd")`	It displays: 7 is odd		
13	`i++`	**3**	7	
14	`i <= 5`	true		
15	`x = Integer.parseInt(cin.nextLine())`	3	**9**	
16	`display(x)`			**9**
17	`if (a % 2 == 0)`			false
18	`System.out.println(a + " is odd")`	It displays: 9 is odd		
19	`i++`	**4**	9	
20	`i <= 5`	true		
21	`x = Integer.parseInt(cin.nextLine())`	4	**2**	
22	`display(x)`			**2**
23	`if (a % 2 == 0)`			true
24	`System.out.println(a + " is even")`	It displays: 2 is even		
25	`i++`	**5**	2	
26	`i <= 5`	true		
27	`x = Integer.parseInt(cin.nextLine())`	5	**4**	
28	`display(x)`			**4**
29	`if (a % 2 == 0)`			true
30	`System.out.println(a + " is even")`	It displays: 4 is even		
31	`i++`	**6**	4	
32	`i <= 5`	false		

6. Solution

Step	Statement	Main Code		void Method `division()`	
		x	y	a	b
1	`x = 20`	**20**	?		
2	`y = 30`	20	**30**		
3	`while (x % y < 30)`	true			

4	`division(y, x)`			**30**	**20**
5	`b = (int)(b / a)`			30	**0**
6	`System.out.println(a * b)`	It displays: 0			
7	`x = 4 * y`	**120**	30		
8	`y++`	120	**31**		
9	`while (x % y < 30)`	true			
10	`division(y, x)`			**31**	**120**
11	`b = (int)(b / a)`			31	**3**
12	`System.out.println(a * b)`	It displays: 93			
13	`x = 4 * y`	**124**	31		
14	`y++`	124	**32**		
15	`while (x % y < 30)`	true			
16	`division(y, x)`			**32**	**124**
17	`b = (int)(b / a)`			32	**3**
18	`System.out.println(a * b)`	It displays: 96			
19	`x = 4 * y`	**128**	32		
20	`y++`	128	**33**		
21	`while (x % y < 30)`	true			
22	`division(y, x)`			**33**	**128**
23	`b = (int)(b / a)`			33	**3**
24	`System.out.println(a * b)`	It displays: 99			
25	`x = 4 * y`	**132**	33		
26	`y++`	132	**34**		
27	`while (x % y < 30)`	false			

7. Solution

Step	Statement	Main Code		void Method `calculate()`		
		i	m	n	s	j
1	`i = 1`	**1**	?			
2	`i <= 3`	true				
3	`m = Integer.parseInt(cin.nextLine())`	1	**2**			
4	`calculate(m)`			**2**	?	?
5	`s = 0`			2	**0**	?
6	`j = 2`			2	0	**2**
7	`j <= 2 * n`			true		
8	`s = s + Math.pow(j, 2)`			2	**4**	2

9	`j += 2`			2	4	**4**
10	`j <= 2 * n`				true	
11	`s = s + Math.pow(j, 2)`			2	**20**	4
12	`j += 2`			2	20	**6**
13	`j <= 2 * n`				false	
14	`System.out.println(s)`	It displays: 20				
15	`i++`	**2**	2			
16	`i <= 3`	true				
17	`m = Integer.parseInt(cin.nextLine())`	2	**3**			
18	`calculate(m)`			**3**	?	?
19	`s = 0`			3	**0**	?
20	`j = 2`			3	0	**2**
21	`j <= 2 * n`				true	
22	`s = s + Math.pow(j, 2)`			3	**4**	2
23	`j += 2`			3	4	**4**
24	`j <= 2 * n`				true	
25	`s = s + Math.pow(j, 2)`			3	**20**	4
26	`j += 2`			3	20	**6**
27	`j <= 2 * n`				true	
28	`s = s + Math.pow(j, 2)`			3	**56**	6
29	`j += 2`			3	56	**8**
30	`j <= 2 * n`				false	
31	`System.out.println(s)`	It displays: 56				
32	`i++`	**3**	3			
33	`i <= 3`	true				
34	`m = Integer.parseInt(cin.nextLine())`	3	**4**			
35	`calculate(m)`			**4**	?	?
36	`s = 0`			4	**0**	?
37	`j = 2`			4	0	**2**
38	`j <= 2 * n`				true	
39	`s = s + Math.pow(j, 2)`			4	**4**	2
40	`j += 2`			4	4	**4**
41	`j <= 2 * n`				true	
42	`s = s + Math.pow(j, 2)`			4	**20**	4
43	`j += 2`			4	20	**6**
44	`j <= 2 * n`				true	
45	`s = s + Math.pow(j, 2)`			4	**56**	6

46	`j += 2`			4	56	**8**
47	`j <= 2 * n`			true		
48	`s = s + Math.pow(j, 2)`			4	**120**	8
49	`j += 2`			4	120	**10**
50	`j <= 2 * n`			false		
51	`System.out.println(s)`	It displays: 120				
52	`i++`	**4**	4			
53	`i <= 3`	false				

8. Solution

```
static double find_sum(double a, double b, double c) {
  return a + b + c;
}
```

9. Solution

```
static double find_avg(double a, double b, double c, double d) {
  return (a + b + c + d) / 4;
}
```

10. Solution

```
static double maximum(double a, double b, double c) {
  double m;

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

11. Solution

```
static void display_max(double a, double b, double c, double d, double e) {
  double m;

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

```
    m = d;
  }
  if (e > m) {
    m = e;
  }
  System.out.println(m);
}
```

12. Solution

```
static double my_round(double x) {
  int digit_to_check;
  double return_value;

  digit_to_check = (int)(x * 1000) % 10;
  if (digit_to_check >= 5) {
    return_value = ((int)(x * 100) + 1) / 100.0;
  }
  else {
    return_value = ((int)(x * 100)) / 100.0;
  }

  return return_value;
}
```

13. Solution

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

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

public static void main(String[] args) {
  double temp1, temp2, x1, x2, x3, x4;

  System.out.print("Enter four numbers: ");
  x1 = Double.parseDouble(cin.nextLine());
  x2 = Double.parseDouble(cin.nextLine());
  x3 = Double.parseDouble(cin.nextLine());
  x4 = Double.parseDouble(cin.nextLine());

  //First approach
  temp1 = find_min(x1, x2);
  temp2 = find_min(x3, x4);
  System.out.println(find_min(temp1, temp2));
```

```
  //Second approach
  System.out.println(find_min(find_min(x1, x2), find_min(x3, x4)));
}
```

14. Solution

```
static double Kelvin_to_Fahrenheit(double kelvin) {
  return 1.8 * kelvin - 459.67;
}

static double Kelvin_to_Celsius(double kelvin) {
  return kelvin - 273.15;
}

public static void main(String[] args) {
  double k;

  System.out.print("Enter a temperature in degrees Kelvin: ");
  k = Double.parseDouble(cin.nextLine());
  System.out.println("Fahrenheit: " + Kelvin_to_Fahrenheit(k));
  System.out.println("Celsius: " + Kelvin_to_Celsius(k));
}
```

15. Solution

```
static String bmi(double w, double h) {
  double b;
  String return_value;

  b = w * 703 / Math.pow(h ,2);
  if (b < 16) {
    return_value = "You must add weight.";
  }
  else if (b < 18.5) {
    return_value = "You should add some weight.";
  }
  else if (b < 25) {
    return_value = "Maintain your weight.";
  }
  else if (b < 30) {
    return_value = "You should lose some weight.";
  }
  else {
    return_value = "You must lose weight.";
  }

  return return_value;
}

public static void main(String[] args) {
  double height, weight;
```

```
  int age;

  System.out.print("Enter your weight (in pounds): ");
  weight = Double.parseDouble(cin.nextLine());
  while (weight < 0) {
    System.out.print("Error! Enter your weight (in pounds): ");
    weight = Double.parseDouble(cin.nextLine());
  }

  System.out.println("Enter your age: ");
  age = Integer.parseInt(cin.nextLine());
  while (age < 18) {
    System.out.print("Error! Enter your age: ");
    age = Integer.parseInt(cin.nextLine());
  }

  System.out.println("Enter your height (in inches): ");
  height = Double.parseDouble(cin.nextLine());
  while (height < 0) {
    System.out.println("Error! Enter your height (in inches): ");
    height = Double.parseDouble(cin.nextLine());
  }

  System.out.println(bmi(weight, height));
}
```

16. Solution

```
static void num_of_days(int year, int month) {
  int days;

  switch (month) {
    case 4:
    case 6:
    case 9:
    case 11:
      days = 30;
      break;
    case 2:
      if (year % 4 == 0 && year % 100 != 0 || year % 400 == 0) {
        days = 29;
      }
      else {
        days = 28;
      }
      break;
    default:
      days = 31;
  }

  System.out.println(days);
```

```
}

public static void main(String[] args) {
  int m, y;

  System.out.print("Enter a year: ");
  y = Integer.parseInt(cin.nextLine());
  for (m = 1; m <= 12; m++) {
    num_of_days(y, m);
  }
}
```

17. Solution

```
static void display_menu() {
  System.out.println();
  System.out.println("1. Convert meters to miles");
  System.out.println("2. Convert miles to meters");
  System.out.println("3. Exit");
  System.out.print("Enter a choice: ");
}

static void meters_to_miles(double meters) {
  System.out.println(meters + " meters equals " + (meters / 1609.344) + " miles");
}

static void miles_to_meters(double miles) {
  System.out.println(miles + " miles equals " + (miles * 1609.344) + " meters");
}

public static void main(String[] args) {
  int choice;
  double distance;

  display_menu();
  choice = Integer.parseInt(cin.nextLine());
  while (choice != 3) {
    System.out.println("Enter distance: ");
    distance = Integer.parseInt(cin.nextLine());
    if (choice == 1) {
      meters_to_miles(distance);
    }
    else {
      miles_to_meters(distance);
    }

    display_menu();
    choice = Integer.parseInt(cin.nextLine());
  }
}
```

18. Solution

```
static void amount_to_pay(int seconds) {
  double extra, tax, total, total_without_tax;

  if (seconds <= 600) {
    extra = 0;
  }
  else if (seconds <= 1200) {
    extra = (seconds - 600) * 0.01;
  }
  else {
    extra = 600 * 0.01 + (seconds - 1200) * 0.02;
  }

  total_without_tax = 10 + extra;
  tax = total_without_tax * 11 / 100;
  total = total_without_tax + tax;

  System.out.println("Total amount to pay: " + total);
}

public static void main(String[] args) {
  int seconds;

  System.out.print("Enter number of seconds: ");
  seconds = Integer.parseInt(cin.nextLine());
  amount_to_pay(seconds);
}
```

Chapter 37

37.9 Review Questions: True/False

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

37.10 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 = Integer.parseInt(cin.nextLine())`	?	?	**12**	?			
2	`arr[1] = 1`	?	**1**	12	?			
3	`arr[0] = 1`	**1**	1	12	?			
4	`while (arr[0] < 8)`	1	1	true				
5	`if (k % arr[1] != 0)`	1	1	false				
6	`x = arr[0] + arr[1] + (int)(arr[0] - arr[1])`	1	1	12	**2**			
7	`System.out.println(arr[1] + " " + arr[0] + " " + x)`	It displays: 1 1 2						
8	`arr[0] += 2`	**3**	1	12	2			
9	`arr[1]++`	3	**2**	12	2			

10	`swap(arr[0], arr[1])`					**3**	**2**	?
11	`temp = x`					3	2	**3**
12	`x = y`					**2**	2	3
13	`y = temp`					2	**3**	3
14	`while (arr[0] < 8)`	**2**	**3**	12	2			
		2	3	true				
15	`if (k % arr[1] != 0)`	2	3	false				
16	`x = arr[0] + arr[1] + (int)(arr[0] - arr[1])`	2	3	12	**4**			
17	`System.out.println(arr[1] + " " + arr[0] + " " + x)`	It displays: 3 2 4						
18	`arr[0] += 2`	**4**	3	12	4			
19	`arr[1]++`	4	**4**	12	4			
20	`swap(arr[0], arr[1])`					**4**	**4**	?
21	`temp = x`					4	4	**4**
22	`x = y`					**4**	4	4
23	`y = temp`					4	**4**	4
24	`while (arr[0] < 8)`	**4**	**4**	12	4			
		4	4	true				
25	`if (k % arr[1] != 0)`	4	4	false				
26	`x = arr[0] + arr[1] + (int)(arr[0] - arr[1])`	4	4	12	**8**			
27	`System.out.println(arr[1] + " " + arr[0] + " " + x)`	It displays: 4 4 8						
28	`arr[0] += 2`	**6**	4	12	8			
29	`arr[1]++`	6	**5**	12	8			
30	`swap(arr[0], arr[1])`					**6**	**5**	?
31	`temp = x`					6	5	**6**
32	`x = y`					**5**	5	6
33	`y = temp`					5	**6**	5
34	`while (arr[0] < 8)`	**5**	**6**	12	8			
		5	6	true				
35	`if (k % arr[1] != 0)`	5	6	false				
36	`x = arr[0] + arr[1] + (int)(arr[0] - arr[1])`	5	6	12	**10**			
37	`System.out.println(arr[1] + " " + arr[0] + " " + x)`	It displays: 6 5 10						
38	`arr[0] += 2`	**7**	6	12	10			
39	`arr[1]++`	7	**7**	12	10			

40	`swap(arr[0], arr[1])`					**7**	**7**	?
41	`temp = x`					7	7	**7**
42	`x = y`					**7**	7	7
43	`y = temp`					7	**7**	7
44	`while (arr[0] < 8)`	**7**	**7**	12	10			
		7	7	true				
45	`if (k % arr[1] != 0)`	7	7	true				
46	`x = arr[0] % arr[1]`	7	7	12	**0**			
47	`swap(arr[1], arr[0])`					**7**	**7**	?
48	`temp = x`					7	7	**7**
49	`x = y`					**7**	7	7
50	`y = temp`					7	**7**	7
51	`System.out.println(arr[1] + " " + arr[0] + " " + x)`	7	7	12	0			
		It displays: 7 7 0						
52	`arr[0] += 2`	**9**	7	12	0			
53	`arr[1]++`	9	**8**	12	0			
54	`swap(arr[0], arr[1])`					**9**	**8**	?
55	`temp = x`					9	8	**9**
56	`x = y`					**8**	8	9
57	`y = temp`					8	**9**	9
58	`while (arr[0] < 8)`	**8**	**9**	12	0			
		false						

5. Solution

It displays: hellohellohello

6. Solution

It displays: 15

7. Solution

It displays: 11 4

8. Solution

```
static final int STUDENTS = 10;
static final int LESSONS = 5;

static void part1(String[] names, int[][] grades) {
  int i, j;

  for (i = 0; i <= STUDENTS - 1; i++) {
```

```
    System.out.print("Enter name for student No. " + (i + 1) + ": ");
    names[i] = cin.nextLine();
    for (j = 0; j <= LESSONS - 1; j++) {
      System.out.print("Enter grade for lesson No. " + (j + 1) + ": ");
      grades[i][j] = Integer.parseInt(cin.nextLine());
    }
  }
}

static double[] part2(int[][] grades) {
  double[] average = new double[STUDENTS];
  int i, j;

  for (i = 0; i <= STUDENTS - 1; i++) {
    average[i] = 0;
    for (j = 0; j <= LESSONS - 1; j++) {
      average[i] += grades[i][j];
    }
    average[i] /= LESSONS;
  }
  return average;
}

static void part3(double[] average, String[] names) {
  int m, n;
  double temp;
  String temp_str;

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

        temp_str = names[n];
        names[n] = names[n - 1];
        names[n - 1] = temp_str;
      }
      else if (average[n] == average[n - 1]) {
        if (names[n].compareTo(names[n - 1]) < 0) {
          temp_str = names[n];
          names[n] = names[n - 1];
          names[n - 1] = temp_str;
        }
      }
    }
  }
}

public static void main(String[] args) {
```

```
  int i;

  String[] names = new String[STUDENTS];
  int[][] grades = new int[STUDENTS][LESSONS];
  double[] average = new double[STUDENTS];

  part1(names, grades);

  average = part2(grades);

  part3(average, names);

  for (i = 0; i <= STUDENTS - 1; i++) {
    System.out.println(names[i] + "\t" + average[i]);
  }
}
```

9. Solution

```
static String part1() {
  String message;

  System.out.print("Enter a message: ");
  message = cin.nextLine().toLowerCase();
  return message;
}

static String part2(String message) {
  String letter, message_clean;
  int i;

  message_clean = "";
  for (i = 0; i <= message.length() - 1; i++) {
    letter = "" + message.charAt(i);
    if (!letter.equals(" ") && !letter.equals(",") &&
        !letter.equals(".") && !letter.equals("?")) {

      message_clean += letter;
    }
  }
  return message_clean;
}

static boolean part3(String message_clean) {
  int middle_pos, i, j;
  boolean palindrome;
  String left_letter, right_letter;

  middle_pos = (int)((message_clean.length() - 1) / 2);
  j = message_clean.length() - 1;
  palindrome = true;
```

```
  for (i = 0; i <= middle_pos; i++) {
    left_letter = "" + message_clean.charAt(i);
    right_letter = "" + message_clean.charAt(j);
    if (!left_letter.equals(right_letter)) {
      palindrome = false;
      break;
    }
    j--;
  }
  return palindrome;
}

static boolean part4(String message) {
  String message_clean;
  boolean palindrome;

  message_clean = part2(message);
  palindrome = part3(message_clean);
  return palindrome;
}

public static void main(String[] args) {
  String message;
  boolean palindrome;

  message = part1();
  palindrome = part4(message);
  if (palindrome) {
    System.out.println("The message is palindrome");
  }
}
```

10. Solution

```
public static void main(String[] args) {
  int a, b, c, maximum;

  a = Integer.parseInt(cin.nextLine());
  b = Integer.parseInt(cin.nextLine());
  c = Integer.parseInt(cin.nextLine());
  d = Integer.parseInt(cin.nextLine());

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

```
  }

  System.out.println(maximum);
}
```

11. Solution

```
static void f1(double a, double b, double c, double[] returning_array) {
  returning_array[0] = a + b + c;
  returning_array[1] = returning_array[0] / 3;
}
```

12. Solution

```
static double my_round(double x, int decimal_places) {
  double return_value;

  int digit_to_check = (int)((x * Math.pow(10, decimal_places + 1))) % 10;
  if (digit_to_check >= 5) {
    return_value = ((int)((x * Math.pow(10, decimal_places))) + 1) / Math.pow(10, decimal_places);
  }
  else {
    return_value = ((int)(x * Math.pow(10, decimal_places))) / Math.pow(10, decimal_places);
  }
  return return_value;
}

static double my_round(double x) {
  return my_round(x, 2);
}
```

13. Solution

```
static boolean get_input() {
  String answer;

  do {
    System.out.print("Enter Yes or No: ");
    answer = cin.nextLine().toUpperCase();
  } while (!answer.equals("YES") && !answer.equals("NO"));

  return answer.equals("YES");  //This returns true or false
}

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

public static void main(String[] args) {
  double b, h;

  do {
```

```
    System.out.print("Enter the base of the parallelogram: ");
    b = Double.parseDouble(cin.nextLine());
    System.out.print("Enter the height of the parallelogram: ");
    h = Double.parseDouble(cin.nextLine());

    System.out.println("Area = " + find_area(b, h));

    System.out.println("Would you like to repeat? ");
  } while (get_input());
}
```

14. Solution

```
static final int STUDENTS = 100;

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

  for (i = 0; i <= STUDENTS - 1; i++) {
    System.out.print("Enter name: ");
    names[i] = cin.nextLine();
    System.out.print("Enter grade: ");
    grades[i] = Integer.parseInt(cin.nextLine());
  }
}

static double get_average(int[] grades) {
  int i, total = 0;
  for (i = 0; i <= STUDENTS - 1; i++) {
    total += grades[i];
  }
  return total / (double)STUDENTS;
}

static void sort_arrays(int[] grades, String[] names) {
  int m, n, element_grds;
  String element_nms;

  for (m = 1; m <= STUDENTS - 1; m++) {
    element_grds = grades[m];
    element_nms = names[m];

    n = m;
    while (n > 0 && grades[n - 1] > element_grds) {
      grades[n] = grades[n - 1];
      names[n] = names[n - 1];
      n--;
    }

    grades[n] = element_grds;
    names[n] = element_nms;
```

```
  }
}

public static void main(String[] args) {
  int i;
  double average;

  String[] names = new String[STUDENTS];
  int[] grades = new int[STUDENTS];

  get_arrays(names, grades);
  average = get_average(grades);
  sort_arrays(grades, names);
  for (i = 0; i <= STUDENTS - 1; i++) {
    if (grades[i] < average) {
      System.out.println(names[i]);
    }
  }
}
```

15. Solution

```
static final int JUDGES = 10;

static int[] get_array() {
  int[] score = new int[JUDGES];
  int i;

  for (i = 0; i <= JUDGES - 1; i++) {
    System.out.print("Judge No " + (i + 1) + ". Enter score: ");
    score[i] = Integer.parseInt(cin.nextLine());
  }
  return score;
}

static void find_min_max(int[] score, int[] results) {
  int i;

  int minimum = score[0];
  int maximum = score[0];
  for (i = 1; i <= JUDGES - 1; i++) {
    if (score[i] > maximum) {
      maximum = score[i];
    }
    if (score[i] < minimum) {
      minimum = score[i];
    }
  }
  results[0] = minimum;
  results[1] = maximum;
}
```

```
public static void main(String[] args) {
  String name;
  int[] results = new int[2];
  int total, i, points;

  System.out.print("Enter artist's name: ");
  name = cin.nextLine();
  int[] score = get_array();
  find_min_max(score, results);

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

  points = total - results[0] - results[1];
  System.out.println("Artist " + name + " got " + points + " points");
}
```

16. Solution

```
static double woc(int index) {
  double return_value;

  if (index == 1) {
    return_value = 1;
  }
  else {
    return_value = 2 * woc(index - 1);
  }
  return return_value;
}

public static void main(String[] args) {
  double total;
  int i;

  total = 0;
  for (i = 1; i <= 64; i++) {
    total += woc(i);
  }
  System.out.println(total);
}
```

17. Solution

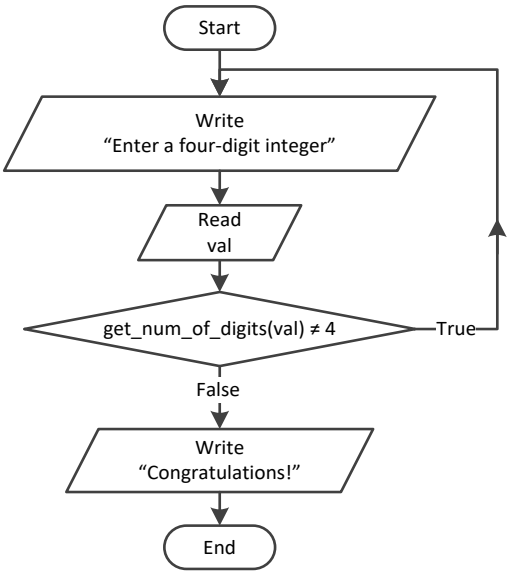
```
static double factorial(int value) {
  double return_value;

  if (value == 1) {
```

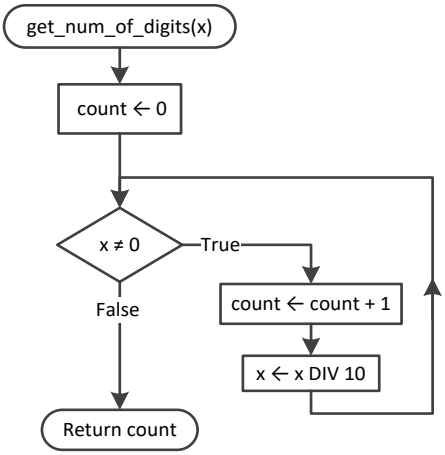
```
    return_value = 1;
  }
  else {
    return_value = value * factorial(value - 1);
  }

  return return_value;
}

static double my_cos(double x, int i) {
  double return_value;

  if (i == 0) {
    return_value = 1;
  }
  else {
    return_value = my_cos(x, i - 4) + Math.pow(x, i) / factorial(i)
                                    - Math.pow(x, i - 2) / factorial(i - 2);
  }

  return return_value;
}

//Overload
static double my_cos(double x) {
  return my_cos(x, 40);
}

public static void main(String[] args) {
  System.out.println(my_cos(Math.PI / 4));
}
```

Chapter 38

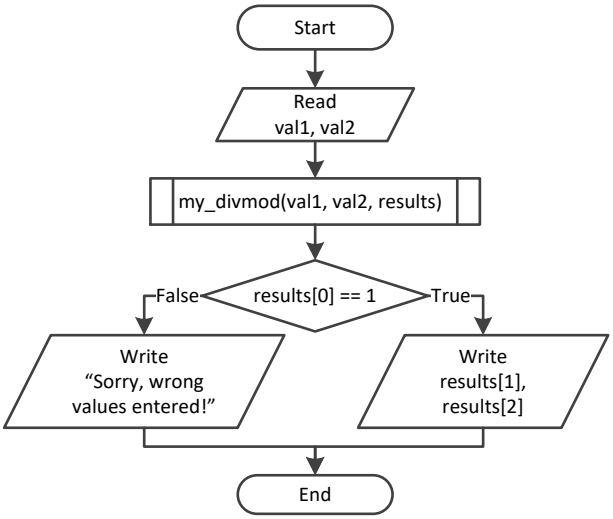
38.3 Review Exercises

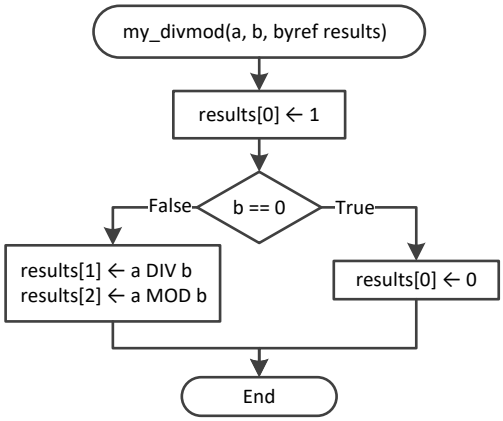
1. Solution





2. Solution

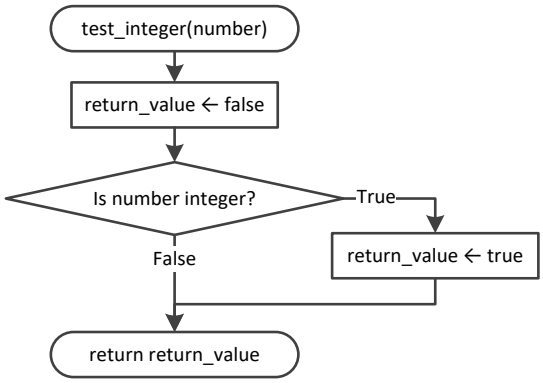


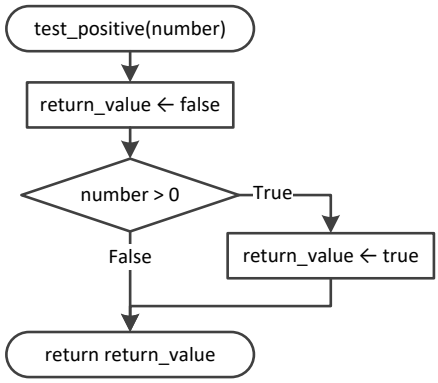


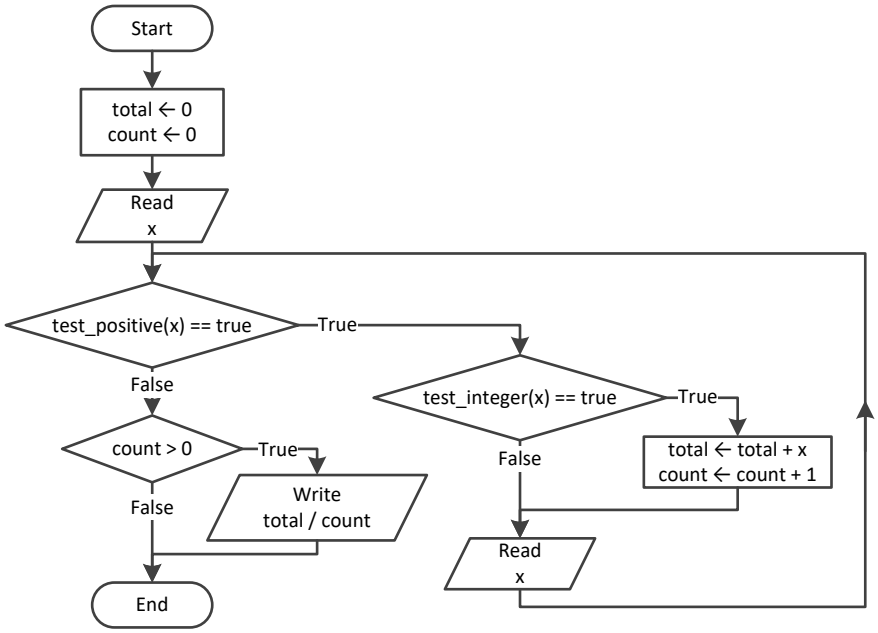
✎ *Flowcharts are a loose method of representing an algorithm. Thus, you can represent a pass by reference using the keyword* `byref`*, which clearly denotes what it actually does.*

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

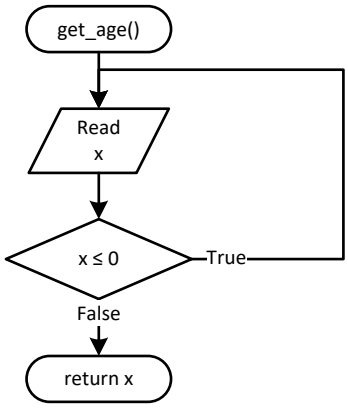
3. Solution

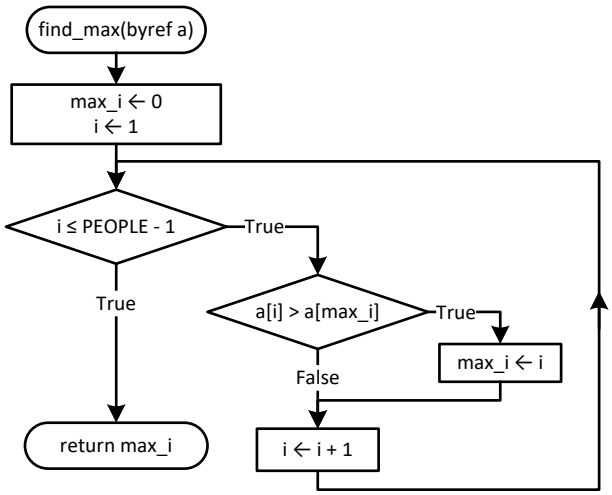

test_integer(number)
return_value ← false
Is number integer?
True
return_value ← true
False
return return_value

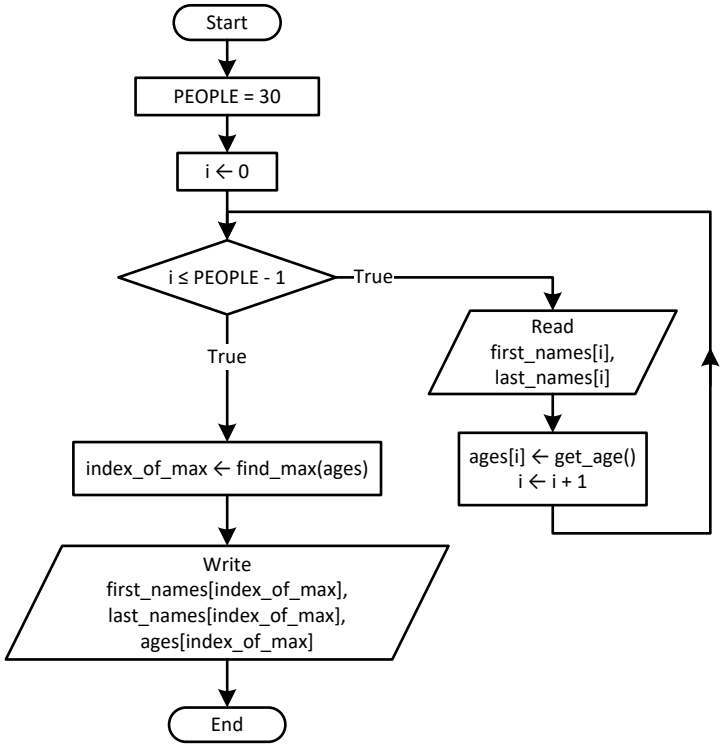

test_positive(number)
return_value ← false
number > 0
True
return_value ← true
False
return return_value


Start
total ← 0
count ← 0
Read
x
test_positive(x) == true
True
False
test_integer(x) == true
True
total ← total + x
count ← count + 1
False
count > 0
True
Write
total / count
False
Read
x
End

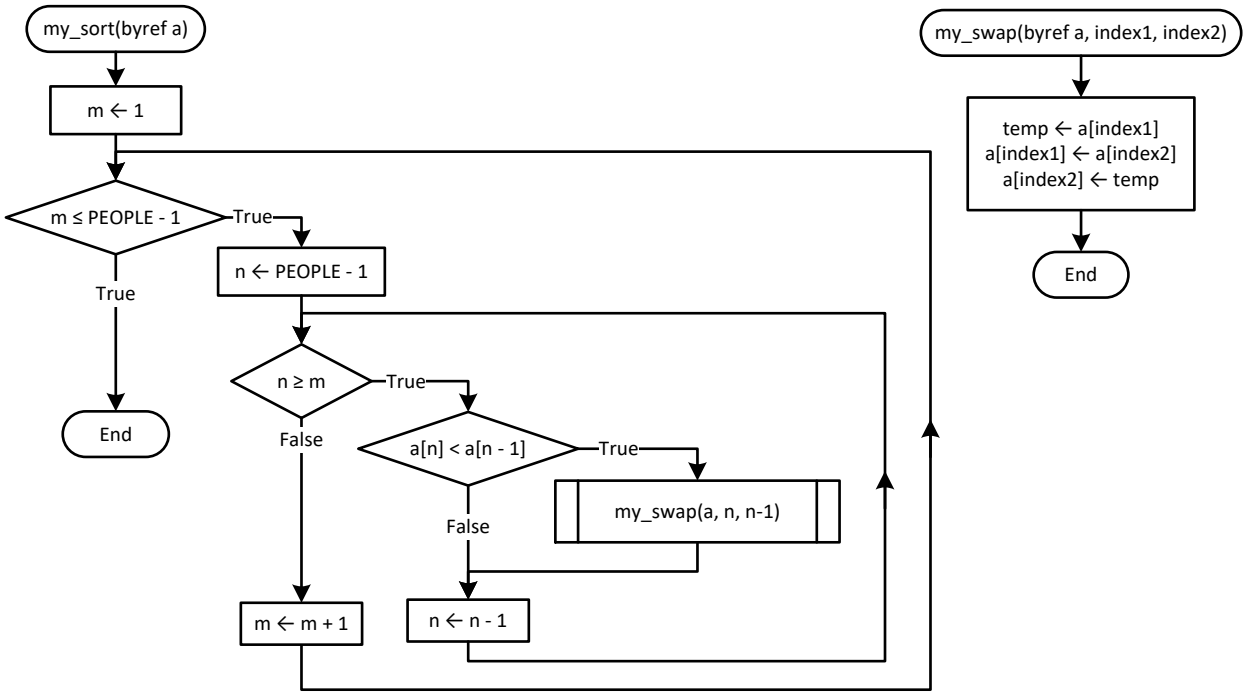
4. Solution

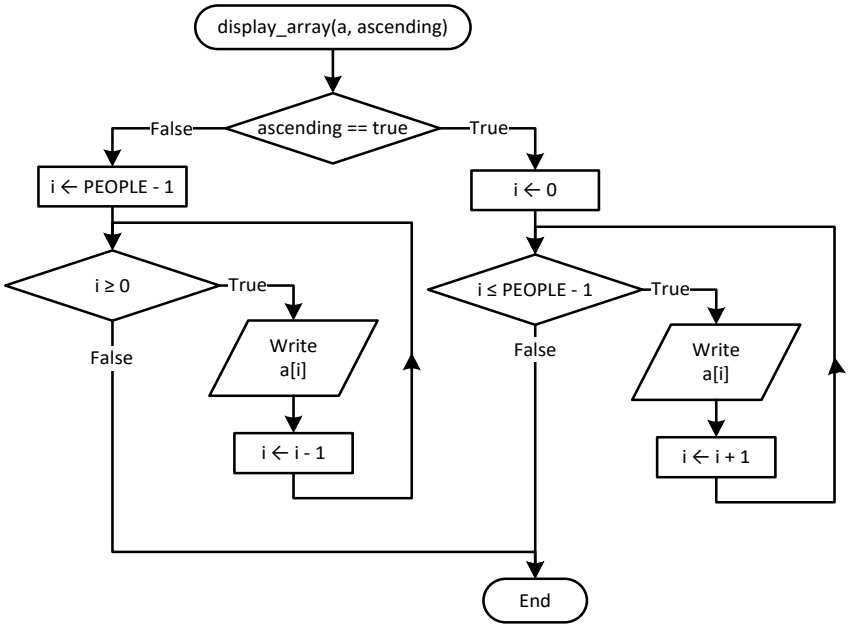

get_age()
Read
x
x ≤ 0
True
False
return x

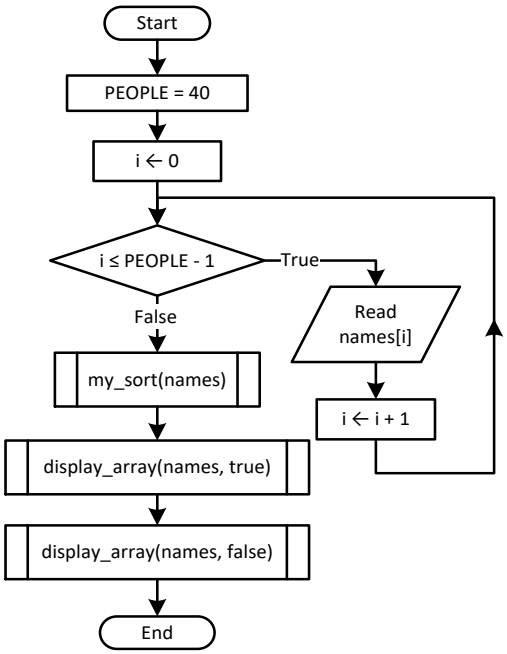

find_max(byref a)
max_i ← 0
i ← 1
i ≤ PEOPLE - 1
True
True
a[i] > a[max_i]
True
max_i ← i
False
i ← i + 1
return max_i


Start
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







6. Solution

```
static double f1(int n) {
  double s;
  int i;

  s = 0;
```

```
  for (i = 1; i <= n; i++) {
    if (i < n / 2.0) {
      s += Math.pow(n, 2);
    }
    else {
      s += Math.pow(n, 3);
    }
  }
  return s;
}

public static void main(String[] args) {
  int val;

  do {
    System.out.print("Enter a positive integer ");
    val = Integer.parseInt(cin.nextLine());
  } while (val < 0);
  System.out.println(f1(val));
}
```

7. Solution

```
static final int ELEMENTS = 100;

static double[] read_values() {
  double[] values = new double[ELEMENTS];
  int i;

  for (i = 0; i <= ELEMENTS - 1; i++) {
    values[i] = Integer.parseInt(cin.nextLine());
  }
  return values;
}

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

  min_i = 0;
  max_i = 0;
  for (i = 1; i <= ELEMENTS - 1; i++) {
    if (values[i] < values[min_i]) {
      min_i = i;
    }
    if (values[i] > values[max_i]) {
      max_i = i;
    }
  }
  results[0] = min_i;
  results[1] = max_i;
}
```

```
public static void main(String[] args) {
  double[] v = new double[ELEMENTS];
  int[] ret = new int[2];

  v = read_values();
  find_min_max(v, ret);
  System.out.println(v[ret[0]] + ", " + v[ret[1]]);
}
```

✎ *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

```
static final double ACCURACY = 0.000000001;

static double factorial(int n) {
  int i;

  double return_value = 1;
  for (i = 1; i <= n; i++) {
    return_value *= i;
  }
  return return_value;
}

static double my_sin(double x) {
  int i, sign;
  double sinus, sinus_previous;
  sign = 1;
  sinus = 0;
  i = 1;
  do {
    sinus_previous = sinus;
    sinus += sign * Math.pow(x, i) / factorial(i);

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

static double degrees_to_rad(double degrees) {
  return 2 * Math.PI * degrees / 360;
}

public static void main(String[] args) {
  int i;
```

```
  for (i = 0; i <= 360; i++) {
    System.out.println("sin(" + i + ") ~= " + my_sin(degrees_to_rad(i)));
  }
}
```

9. Solution

```
static boolean is_leap(int year) {
  boolean return_value = false;
  if (year % 4 == 0 && year % 100 != 0 || year % 400 == 0) {
    return_value = true;
  }
  return return_value;
}

static int num_of_days(int year, int month) {
  int days;

  switch (month) {
    case 4:
    case 6:
    case 9:
    case 11:
      days = 30;
      break;
    case 2:
      if (is_leap(year)) {
        days = 29;
      }
      else {
        days = 28;
      }
      break;
    default:
      days = 31;
  }

  return days;
}

static boolean check_date(int day, int month, int year) {
  boolean return_value = true;
  if (month < 1 || month > 12) {
    return_value = false;
  }
  else if (day < 1 || day > num_of_days(year, month)) {
    return_value = false;
  }
  return return_value;
}
```

```
public static void main(String[] args) {
  int day, month, year, total, i;

  System.out.print("Enter day: ");
  day = Integer.parseInt(cin.nextLine());
  System.out.print("Enter month: ");
  month = Integer.parseInt(cin.nextLine());
  System.out.print("Enter year: ");
  year = Integer.parseInt(cin.nextLine());
  while (!check_date(day, month, year)) {
    System.out.println("Error!");
    System.out.print("Enter day: ");
    day = Integer.parseInt(cin.nextLine());
    System.out.print("Enter month: ");
    month = Integer.parseInt(cin.nextLine());
    System.out.print("Enter year: ");
    year = Integer.parseInt(cin.nextLine());
  }

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

  System.out.println(total);
}
```

10. Solution

```
static void display_menu() {
  System.out.println("----------------------------------------------");
  System.out.println("1. Convert USD to Euro (EUR)");
  System.out.println("2. Convert USD to British Pound Sterling (GBP)");
  System.out.println("3. Convert EUR to USD");
  System.out.println("4. Convert EUR to GBP");
  System.out.println("5. Convert GBP to USD");
  System.out.println("6. Convert GBP to EUR");
  System.out.println("7. Exit");
  System.out.println("-------------------------");
  System.out.print("Enter a choice: ");
}

static double USD_to_EUR(double value) {
  return value * 0.87;
}

static double USD_to_GBP(double value) {
  return value * 0.76;
}
```

```
public static void main(String[] args) {
  int choice;
  double amount;

  display_menu();
  choice = Integer.parseInt(cin.nextLine());
  while (choice != 7) {
    System.out.print("Enter an amount: ");
    amount = Double.parseDouble(cin.nextLine());
    switch (choice) {
      case 1:
        System.out.println(amount + " USD = " + USD_to_EUR(amount) + " Euro");
        break;
      case 2:
        System.out.println(amount + " USD = " + USD_to_GBP(amount) + " GBP");
        break;
      case 3:
        System.out.println(amount + " EUR = " + 1 / USD_to_EUR(1 / amount) + " USD");
        break;
      case 4:
        System.out.println(amount + " EUR = " + USD_to_GBP(1 / USD_to_EUR(1 / amount)) + " GBP");
        break;
      case 5:
        System.out.println(amount + " GBP = " + 1 / USD_to_GBP(1 / amount) + " USD");
        break;
      case 6:
        System.out.println(amount + " GBP = " + USD_to_EUR(1 / USD_to_GBP(1 / amount)) + " EUR");
        break;
    }

    display_menu();
    choice = Integer.parseInt(cin.nextLine());
  }
}
```

11. Solution

```
static int dice() {
  return 1 + (int)(Math.random() * 6);
}

public static void main(String[] args) {
  int dice1, dice2, i, player, total, total_player1 = 0, total_player2 = 0;
  String[] names = new String[2];

  System.out.print("Player1 - Enter name: ");
  names[0] = cin.nextLine();
  System.out.print("Player2 - Enter name: ");
  names[1] = cin.nextLine();

  for (player = 0; player <= 1; player++) {
```

```
    total = 0;
    for (i = 1; i <= 10; i++) {
      System.out.println(names[player] + ", hit enter to roll the dice!");
      cin.nextLine();  //This statement just waits the user to hit the enter key

      dice1 =  dice();
      dice2 =  dice();
      System.out.println(dice1 + "   " + dice2);
      total += dice1 + dice2;
    }
    if (player == 1) {
      total_player1 = total;
    }
    else {
      total_player2 = total;
    }
  }

  if (total_player1 == total_player2) {
    System.out.println("Tie!");
  }
  else if (total_player1 > total_player2) {
    System.out.println(names[0] + " wins");
  }
  else {
    System.out.println(names[1] + " wins");
  }
}
```

12. Solution

```
static final int GAS = 1;
static final int DIESEL = 2;
static final int HYBRID = 3;
static final double TAX_RATE = 0.10;
static final int CARS = 40;

static int get_choice() {
  System.out.println("1. Gas");
  System.out.println("2. Diesel");
  System.out.println("3. Hybrid");
  System.out.print("Enter type of the car: ");
  return Integer.parseInt(cin.nextLine());
}

static int get_days() {
  System.out.print("Enter total number of rental days: ");
  return Integer.parseInt(cin.nextLine());
}

static double get_charge(int car_type, int rental_days) {
```

```
  double charge;

  if (car_type == GAS) {
    if (rental_days <= 5) {
      charge = rental_days * 24;
    }
    else if (rental_days <= 8) {
      charge = 5 * 24 + (rental_days - 5) * 22;
    }
    else {
      charge = 5 * 24 + 3 * 22 + (rental_days - 8) * 18;
    }
  }
  else if (car_type == DIESEL) {
    if (rental_days <= 5) {
      charge = rental_days * 28;
    }
    else if (rental_days <= 8) {
      charge = 5 * 28 + (rental_days - 5) * 25;
    }
    else {
      charge = 5 * 28 + 3 * 25 + (rental_days - 8) * 21;
    }
  }
  else {
    if (rental_days <= 5) {
      charge = rental_days * 30;
    }
    else if (rental_days <= 8) {
      charge = 5 * 30 + (rental_days - 5) * 28;
    }
    else {
      charge = 5 * 30 + 3 * 28 + (rental_days - 8) * 23;
    }
  }
  charge = charge * (1 + TAX_RATE); //This is equivalent to charge += charge * TAX_RATE;
  return charge;
}

public static void main(String[] args) {
  int count, i;
  double charge, total;

  int[] rented_car_types = new int[CARS];
  int[] rented_days = new int[CARS];

  for (i = 0; i <= CARS - 1; i++) {
    rented_car_types[i] = get_choice();
    rented_days[i] = get_days();
  }
```

```
  total = 0;
  for (i = 0; i <= CARS - 1; i++) {
    charge = get_charge(rented_car_types[i], rented_days[i]);
    System.out.println("Car No " + (i + 1) + ": " + charge);
    total += charge;
  }

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

  System.out.println("Hybrids rented: " + count);
  System.out.println("Net profit: " + total / (1 + TAX_RATE));
}
```

13. Solution

```
static final int CHANNELS = 10;
static final int DAYS = 7;
static final String day_names[] = {"Monday", "Tuesday", "Wednesday",
                                   "Thursday", "Friday", "Saturday", "Sunday"};

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

  for (i = 0; i <= CHANNELS - 1; i++) {
    System.out.println("Enter name for channel No. " + (i + 1) + ": ");
    names[i] = cin.nextLine();
    for (j = 0; j <= DAYS - 1; j++) {
      System.out.print("Enter the number of viewers of the main news program on " + day_names[j]);
      System.out.print(" for channel " + names[i] + ": ");
      viewers[i][j] = Integer.parseInt(cin.nextLine());
    }
  }
}

static double get_average(int a[]) {
  int total ,i;

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

public static void main(String[] args) {
  int i, j;
```

```
  double weekend;
  boolean increasing;

  String[] names = new String[CHANNELS];
  int[][] viewers = new int[CHANNELS][DAYS];
  get_data(names, viewers);

  int[] temporary_array = new int[5];
  for (i = 0; i <= CHANNELS - 1; i++) {
    for (j = 0; j <= 4; j++) {
      temporary_array[j] = viewers[i][j];
    }
    weekend = (viewers[i][DAYS - 2] + viewers[i][DAYS - 1]) / 2;
    if (weekend >= 1.2 * get_average(temporary_array)) {
      System.out.println(names[i]);
    }
  }

  for (i = 0; i <= CHANNELS - 1; i++) {
    increasing = true;
    for (j = 1; j <= DAYS - 1; j++) {
      if (viewers[i][j] <= viewers[i][j - 1]) {
        increasing = false;
      }
    }
    if (increasing) {
      System.out.println(names[i]);
    }
  }
}
```

14. Solution

```
static final int CITIZENS = 300;

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

  for (i = 0; i <= CITIZENS - 1; i++) {
    System.out.print("Enter SSN: ");
    SSNs[i] = Long.parseLong(cin.nextLine());
    System.out.print("Enter answer: ");
    answers[i] = cin.nextLine();
  }
}

static void sort_arrays(long[] SSNs, String[] answers) {
  int m, n, index_of_min;
  long minimum, temp;
  String temp_str;
```

```
  for (m = 0; m <= CITIZENS - 1; m++) {
    minimum = SSNs[m];
    index_of_min = m;
    for (n = m; n <= CITIZENS - 1; n++) {
      if (SSNs[n] < minimum) {
        minimum = SSNs[n];
        index_of_min = n;
      }
    }
    temp = SSNs[m];
    SSNs[m] = SSNs[index_of_min];
    SSNs[index_of_min] = temp;
    temp_str = answers[m];
    answers[m] = answers[index_of_min];
    answers[index_of_min] = temp_str;
  }
}

static int search_array(long[] SSNs, long SSN) {
  int left, right, middle, index_position = 0, return_value;
  boolean found;

  left = 0;
  right = CITIZENS - 1;
  found = false;
  while (left <= right && !found) {
    middle = (int)((left + right) / 2);

    if (SSNs[middle] > SSN) {
      right = middle - 1;
    }
    else if (SSNs[middle] < SSN) {
      left = middle + 1;
    }
    else {
      found = true;
      index_position = middle;
    }
  }

  if (!found) {
    System.out.println("SSN not found!");
    return_value = -1;
  }
  else {
    return_value = index_position;
  }
  return return_value;
}

static int count_answers(String[] answers, String answer) {
```

```
  int count, i;

  count = 0;
  for (i = 0; i <= CITIZENS - 1; i++) {
    if (answers[i].equals(answer)) {
      count++;
    }
  }
  return count;
}

public static void main(String[] args) {
  long[] SSNs = new long[CITIZENS];
  long SSN;
  String[] answers = new String[CITIZENS];
  int index, count;
  String answer;

  do {
    input_data(SSNs, answers);
    sort_arrays(SSNs, answers);

    System.out.print("Enter an SSN to search: ");
    SSN = Long.parseLong(cin.nextLine());

    index = search_array(SSNs, SSN);
    if (index != -1) {
      answer = answers[index];
      System.out.println(answer);

      count = count_answers(answers, answer);
      System.out.println(count * 100 / (double)CITIZENS);
    }
    System.out.println("Repeat? ");
    answer = cin.nextLine();
  } while (answer.equals("yes"));
}
```

15. Solution

```
static final int TEAMS = 8;
static final int GAMES = 12;

static void input_data(String[] names, String[][] results)  {
  int i, j;

  for (i = 0; i <= TEAMS - 1; i++) {
    System.out.print("Enter team name: ");
    names[i] = cin.nextLine();
    for (j = 0; j <= GAMES - 1; j++) {
      System.out.print("Enter result (W, L, T): ");
```

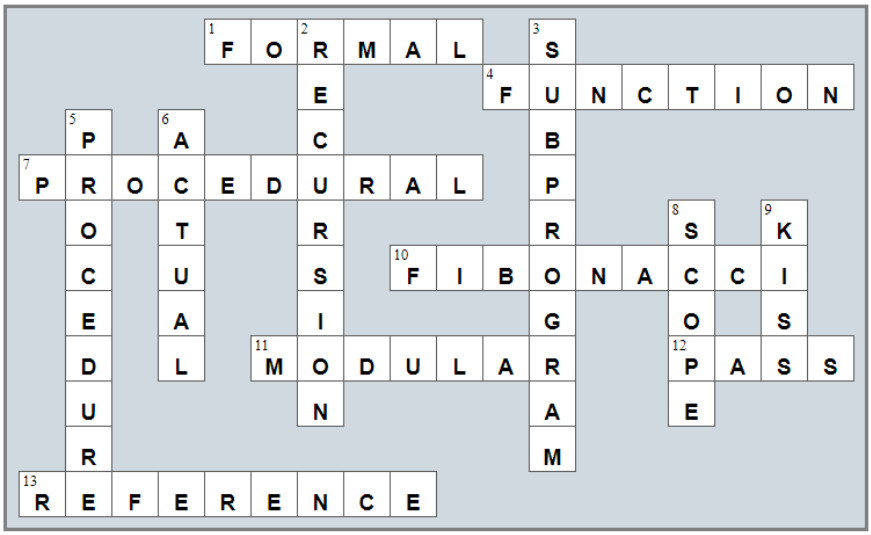
```
      results[i][j] = cin.nextLine();
    }
  }
}

static void display_result(String[] names, String[][] results) {
  String result;
  int i, j;
  boolean found;

  System.out.print("Enter a result to search (W, L, T): ");
  result = cin.nextLine();
  for (i = 0; i <= TEAMS - 1; i++) {
    System.out.println("Team: " + names[i]);
    found = false;
    for (j = 0; j <= GAMES - 1; j++) {
      if (results[i][j].equals(result)) {
        System.out.println("Week: " + (j + 1));
        found = true;
      }
    }
    if (!found) {
      System.out.println("Nothing found");
    }
  }
}

static int find_team(String[] names) {
  String name;
  int i, return_value;

  System.out.print("Enter a name to search: ");
  name = cin.nextLine();

  i = 0;
  while (i < TEAMS - 1 && !names[i].equals(name)) {
    i++;
  }

  if (!names[i].equals(name)) {
    return_value = -1;
  }
  else {
    return_value = i;
  }
  return return_value;
}

public static void main(String[] args) {
  String[] names = new String[TEAMS];
  String[][] results = new String[TEAMS][GAMES];
```

```
  int j, index, total;

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

  index = find_team(names);
  while (index != -1) {
    total = 0;
    for (j = 0; j <= GAMES - 1; j++) {
      if (results[index][j].equals("W")) {
        total += 3;
      }
      else if (results[index][j].equals("T")) {
        total += 1;
      }
    }
    System.out.println("Points: " + total);
    index = find_team(names);
  }

  if (index == -1) {
    System.out.println("Team not found");
  }
}
```

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 double square_area(double side) {
    return side * side;
  }

  public double rectangle_area(double b, double h) {
    return b * h;
  }

  public double triangle_area(double b, double h) {
    return b * h / 2;
  }
}

public class MainClass {
  static Scanner cin = new Scanner(System.in);

  public static void main(String[] args) {
    double sqr_side, rctngl_base, rctngl_height, trngl_base, trngl_height;
    Trigonometry tr = new Trigonometry();

    System.out.print("Enter square side: ");
    sqr_side = Double.parseDouble(cin.nextLine());

    System.out.print("Enter rectangle base:  ");
    rctngl_base = Double.parseDouble(cin.nextLine());
    System.out.print("Enter rectangle height: ");
    rctngl_height = Double.parseDouble(cin.nextLine());

    System.out.print("Enter triangle base: ");
    trngl_base = Double.parseDouble(cin.nextLine());
    System.out.print("Enter triangle height: ");
    trngl_height = Double.parseDouble(cin.nextLine());

    System.out.println(tr.square_area(sqr_side));
    System.out.println(tr.rectangle_area(rctngl_base, rctngl_height));
```

```
    System.out.println(tr.triangle_area(trngl_base, trngl_height));
  }
}
```

2. Solution

```
class Pet {
  public String kind;
  public int legs_number;

  public void start_running() {
    System.out.println("Pet is running");
  }

  public void stop_running() {
    System.out.println("Pet stopped");
  }
}

public class MainClass {
  static Scanner cin = new Scanner(System.in);

  public static void main(String[] args) {
    Pet pet1 = new Pet();
    pet1.kind = "dog";
    pet1.legs_number = 4;

    Pet pet2 = new Pet();
    pet2.kind = "monkey";
    pet2.legs_number = 2;

    pet1.start_running();
    pet2.start_running();
    pet1.stop_running();
  }
}
```

3. Solution

```
class Pet {
  private String kind;
  private int legs_number;

  //Define the constructor
  public Pet(String kind, int legs_number) throws Exception {
    this.setKind(kind);
    this.setLegs_number(legs_number);
  }

  public String getKind() {
    return this.kind;
```

```
  }

  public void setKind(String value) throws Exception {
    if (!value.equals("")) {
      this.kind = value;
    }
    else {
      throw new Exception("Cannot be empty");
    }
  }

  public int getLegs_number() {
    return this.legs_number;
  }

  public void setLegs_number(int value) throws Exception {
    if (value >= 0) {
      this.legs_number = value;
    }
    else {
      throw new Exception("Cannot be negative");
    }
  }

  public void start_running() {
    System.out.println("Pet is running");
  }

  public void stop_running() {
    System.out.println("Pet stopped");
  }
}

public class MainClass {
  static Scanner cin = new Scanner(System.in);

  public static void main(String[] args) throws Exception {
    Pet pet1 = new Pet("dog", 4);

    pet1.start_running();
    pet1.stop_running();

    pet1.setKind(""); //This will throw an error
    pet1.setLegs_number(-1); //This will throw an error
  }
}
```

4. Solution

```
class Box {
  private double width;
```

```
  private double length;
  private double height;

  //Define the constructor
  public Box(double w, double l, double h) {
    //Initialize fields
    this.width = w;
    this.length = l;
    this.height = h;
  }

  public void display_volume() {
    System.out.println("Volume: " + (this.width * this.length * this.height));
  }

  public void display_dimensions() {
    System.out.println(this.width + " x " + this.length + " x " + this.height);
  }
}

public class MainClass {
  static Scanner cin = new Scanner(System.in);
  static final int BOXES = 3;

  public static void main(String[] args) {
    int i;
    double w, l, h;

    Box[] list_of_obj = new Box[BOXES]; //create an array

    for (i = 0; i <= BOXES - 1; i++) {
      System.out.print("Enter width: ");
        w = Double.parseDouble(cin.nextLine());
        System.out.print("Enter length: ");
        l = Double.parseDouble(cin.nextLine());
        System.out.print("Enter height: ");
        h = Double.parseDouble(cin.nextLine());

        //Add each new object to the array
        list_of_obj[i] = new Box(w, l, h);
    }

    for (i = 0; i <= BOXES - 1; i++) {
        list_of_obj[i].display_dimensions();
        list_of_obj[i].display_volume();
    }
  }
}
```

5. Solution

```
class Box {
  private double width;
  private double length;
  private double height;

  //Define the constructor
  public Box(double w, double l, double h) throws Exception {
      //Initialize fields (using the corresponding set methods)
      this.setWidth(w);
      this.setLength(l);
      this.setHeight(h);
  }

  //Define the getter
  public double getWidth() {
    return this.width;
  }

  //Define the setter
  public void setWidth(double value) throws Exception {
    if (value > 0) {
      this.width = value;
    }
    else {
      throw new Exception("Cannot be negative or zero");
    }
  }

  //Define the getter
  public double getLength() {
    return this.length;
  }

  //Define the setter
  public void setLength(double value) throws Exception {
    if (value > 0) {
      this.length = value;
    }
    else {
      throw new Exception("Cannot be negative or zero");
    }
  }

  //Define the getter
  public double getHeight() {
    return this.height;
  }

  //Define the setter
```

```
  public void setHeight(double value) throws Exception {
    if (value > 0) {
      this.height = value;
    }
    else {
      throw new Exception("Cannot be negative or zero");
    }
  }

  public void display_volume() {
    System.out.println("Volume: " + (this.width * this.length * this.height));
  }

  public void display_dimensions() {
    System.out.println(this.width + " x " + this.length + " x " + this.height);
  }
}

public class MainClass {
  static Scanner cin = new Scanner(System.in);
  static final int BOXES = 3;

  public static void main(String[] args) throws Exception {
    int i;
    double w, l, h;

    Box[] list_of_obj = new Box[BOXES]; //Create an array

    for (i = 0; i <= BOXES - 1; i++) {
      System.out.print("Enter width: ");
        w = Double.parseDouble(cin.nextLine());
        System.out.print("Enter length: ");
        l = Double.parseDouble(cin.nextLine());
        System.out.print("Enter height: ");
        h = Double.parseDouble(cin.nextLine());

        //Add each new object to the array
        list_of_obj[i] = new Box(w, l, h);
    }

    for (i = 0; i <= BOXES - 1; i++) {
        list_of_obj[i].display_dimensions();
        list_of_obj[i].display_volume();
    }
  }
}
```

6. Solution

```
class Cube {
  private double edge;
```

```
  //Define the constructor
  public Cube(double edge) {
    this.edge = edge;
  }

  public void display_volume() {
    System.out.println("Volume: " + Math.pow(this.edge, 3));
  }

  public void display_one_surface() {
    System.out.println("One surface: " + Math.pow(this.edge, 2));
  }

  public void display_total_surface() {
    System.out.println("Total surface: " + 6 * Math.pow(this.edge, 2));
  }
}

public class MainClass {
  static Scanner cin = new Scanner(System.in);

  public static void main(String[] args) {
    double edge;

    System.out.print("Enter edge length of a cube: ");
    edge = Double.parseDouble(cin.nextLine());

    Cube cube1 = new Cube(edge);

    cube1.display_volume();
    cube1.display_one_surface();
    cube1.display_total_surface();
  }
}
```

7. Solution

```
class Cube {
  private double edge;

  //Define the constructor
  public Cube(double edge) throws Exception {
    this.setEdge(edge);
  }

  //Define the getter
  public double getEdge() {
    return this.edge;
  }
```

```
  //Define the setter
  public void setEdge(double value) throws Exception {
    if (value > 0) {
      this.edge = value;
    }
    else {
      throw new Exception("Cannot be negative or zero");
    }
  }

  public void display_volume() {
    System.out.println("Volume: " + Math.pow(this.edge, 3));
  }

  public void display_one_surface() {
    System.out.println("One surface: " + Math.pow(this.edge, 2));
  }

  public void display_total_surface() {
    System.out.println("Total surface: " + 6 * Math.pow(this.edge, 2));
  }
}

public class MainClass {
  static Scanner cin = new Scanner(System.in);

  public static void main(String[] args) throws Exception {
    double edge;

    System.out.print("Enter edge length of a cube: ");
    edge = Double.parseDouble(cin.nextLine());

    Cube cube1 = new Cube(edge);

    cube1.display_volume();
    cube1.display_one_surface();
    cube1.display_total_surface();
  }
}
```

8. Solution

```
class Circle {
  private double radius = -1;

  //Define the getter
  public double getRadius() throws Exception {
    if (this.radius > 0) {
      return this.radius;
    }
    else {
```

```
      throw new Exception("Radius is not set");
    }
  }

  //Define the setter
  public void setRadius(double value) throws Exception {
    if (value > 0) {
      this.radius = value;
    }
    else {
      throw new Exception("Cannot be negative or zero");
    }
  }

  public double get_diameter() throws Exception {
    return 2 * this.getRadius();
  }

  public double get_area() throws Exception {
    return 3.14 * Math.pow(this.getRadius(), 2);
  }

  public double get_perimeter() throws Exception {
      return 2 * 3.14 * this.getRadius();
  }
}

public class MainClass {
  static Scanner cin = new Scanner(System.in);

  static void display_menu() {
    System.out.println("1. Enter radius");
    System.out.println("2. Display radius");
    System.out.println("3. Display diameter");
    System.out.println("4. Display area");
    System.out.println("5. Display perimeter");
    System.out.println("6. Exit");
  }

  public static void main(String[] args) throws Exception {
    int choice;
    double radius;

    Circle circle1 = new Circle();

    while (true) {
      display_menu();
      System.out.print("Enter a choice: ");
      choice = Integer.parseInt(cin.nextLine());

      if (choice == 6) {
```

```
        System.out.println("Bye");
        break;
      }
      else if (choice == 1) {
        System.out.print("Enter radius: ");
        radius = Double.parseDouble(cin.nextLine());
        circle1.setRadius(radius);
      }
      else if (choice == 2) {
        System.out.println("Radius: " + circle1.getRadius());
      }
      else if (choice == 3) {
        System.out.println("Diameter: " + circle1.get_diameter());
      }
      else if (choice == 4) {
        System.out.println("Area: " + circle1.get_area());
      }
      else if (choice == 5) {
        System.out.println("Perimeter: " + circle1.get_perimeter());
      }
    }
  }
}
```

9. Solution

```
class Info {
  private String user_text;

  //Define the getter
  public String getUser_text() {
    return this.user_text;
  }

  //Define the setter
  public void setUser_text(String value) throws Exception {
    if (!value.equals("")) {
      this.user_text = value;
    }
    else {
      throw new Exception("Cannot be set to empty");
    }
  }

  public int get_spaces_count() {
      int i, count = 0;
      String character;

      for (i = 0; i <= this.getUser_text().length() - 1; i++) {
        character = "" + this.getUser_text().charAt(i);
        if (character.equals(" ")) {
```

```
            count += 1;
          }
        }
        return count;
    }

    public int get_words_count() {
        return this.get_spaces_count() + 1;
    }

    public int get_vowels_count() {
      int i, count = 0;
      String character;

      for (i = 0; i <= this.getUser_text().length() - 1; i++) {
        character = "" + this.getUser_text().toLowerCase().charAt(i);
        if ("aeiou".indexOf(character) > -1) {
          count += 1;
        }
      }
      return count;
    }

    public int get_letters_count() {
      return this.getUser_text().length() - this.get_spaces_count();
    }
  }

  public class classMain {
    static Scanner cin = new Scanner(System.in);

    public static void main(String[] args) throws Exception {
      Info inf = new Info();

      System.out.print("Enter a text: ");
      inf.setUser_text(cin.nextLine());

      System.out.println("Text: " + inf.getUser_text());
      System.out.println("Spaces: " + inf.get_spaces_count());
      System.out.println("Words: " + inf.get_words_count());
      System.out.println("Vowels: " + inf.get_vowels_count());
      System.out.println("Total number of letters: " + inf.get_letters_count());
    }
  }
```

10. Solution

```
class EncryptDecrypt {
  static final String alphabet = " abcdefghijklmnopqrstuvwxyz";  //space is a valid character!
  private int encr_decr_key = -1;
```

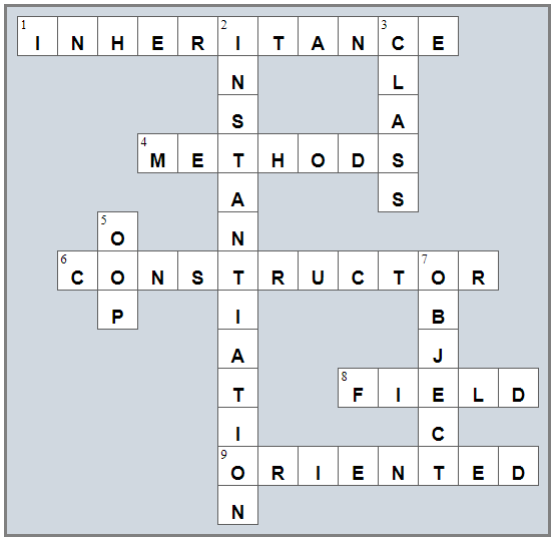
```
//Define the getter
public int getEncr_decr_key() throws Exception {
  if (this.encr_decr_key != -1) {
    return this.encr_decr_key;
  }
  else {
    throw new Exception("Key is not set");
  }
}

//Define the setter
public void setEncr_decr_key(int value) throws Exception {
  if (value >= 1 && value <= 26) {
    this.encr_decr_key = value;
  }
  else {
    throw new Exception("Must be between 1 and 26");
  }
}

public String encrypt(String message) throws Exception {
  String character, return_value = "";
  char new_letter;
  int i, index, new_index;

  for (i = 0; i <= message.length() - 1; i++) {
    character = "" + message.charAt(i);
    index = alphabet.indexOf(character);
    new_index = index + this.getEncr_decr_key();
    if (new_index >= 27) {
      new_index -= 27;
    }
    new_letter = alphabet.charAt(new_index);
    return_value += new_letter;
  }
  return return_value;
}

public String decrypt(String enc_message) throws Exception {
  String character, return_value = "";
  char new_letter;
  int i, index, new_index;

  for (i = 0; i <= enc_message.length() - 1; i++) {
    character = "" + enc_message.charAt(i);
    index = alphabet.indexOf(character);
    new_index = index - this.getEncr_decr_key();
    if (new_index < 0) {
      new_index += 27;
    }
    new_letter = alphabet.charAt(new_index);
```

```
      return_value += new_letter;
    }
    return return_value;
  }
}

public class MainClass {
  static Scanner cin = new Scanner(System.in);

  static void display_menu() {
    System.out.println("1. Enter encryption/decryption key");
    System.out.println("2. Encrypt a message");
    System.out.println("3. Decrypt a message");
    System.out.println("4. Exit");
  }

  public static void main(String[] args) throws Exception {
    String text;
    int choice;

    EncryptDecrypt ed = new EncryptDecrypt();

    display_menu();
    System.out.print("Enter a choice: ");
    choice = Integer.parseInt(cin.nextLine());
    while (choice != 4) {
      if (choice == 1) {
        System.out.print("Enter encryption/decryption key: ");
        ed.setEncr_decr_key(Integer.parseInt(cin.nextLine()));
      }
      else if (choice == 2) {
        System.out.print("Enter message to encrypt: ");
        text = cin.nextLine();
        System.out.println("Encrypted message: " + ed.encrypt(text));
      }
      else if (choice == 3) {
        System.out.print("Enter message to decrypt: ");
        text = cin.nextLine();
        System.out.println("Decrypted message: " + ed.decrypt(text));
      }

      display_menu();
      System.out.print("Enter a choice: ");
      choice = Integer.parseInt(cin.nextLine());
    }
  }
}
```

Review in “Object Oriented Programming”

Review Crossword Puzzle

1.



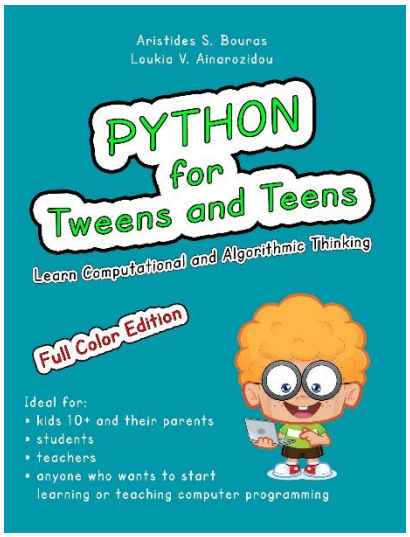
Some Final Words from the Author

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.

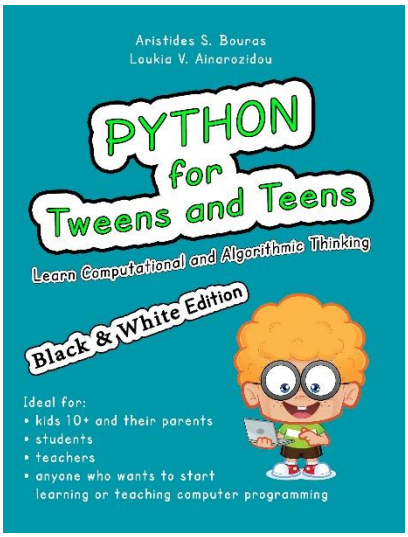
So if you liked this book, please visit the web store where you bought it and show me your gratitude by writing a good review and giving me as many stars as possible. By doing this, you will encourage me to continue writing and of course you'll help other readers to reach me.

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

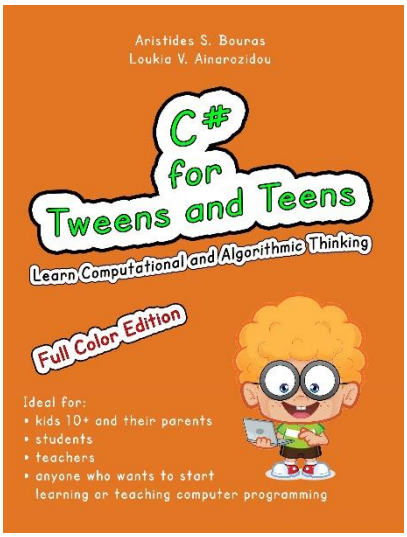
Some of my Books



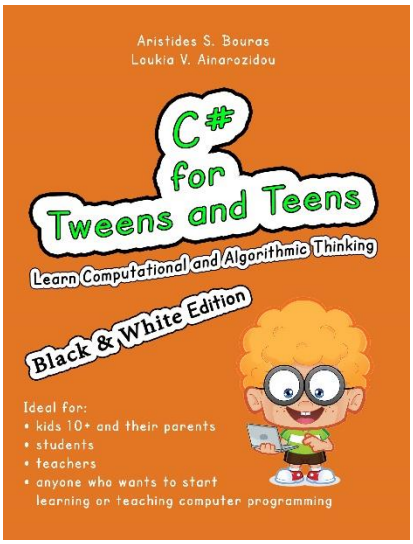
ISBN-10: 1543127940
ISBN-13: 978-1543127942



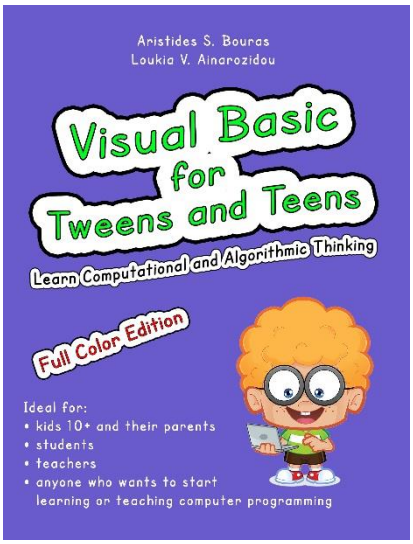
ISBN-10: 1546611215
ISBN-13: 978-1546611219



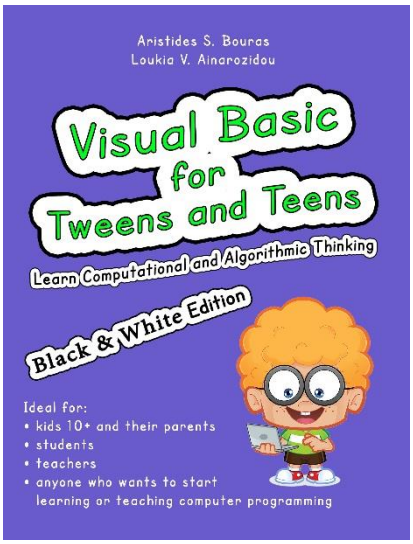
ISBN-10: 1973727684
ISBN-13: 978-1973727682



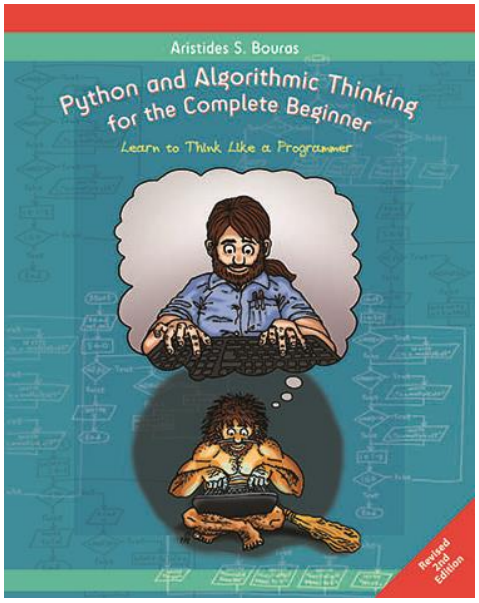
ISBN-10: 1973727765
ISBN-13: 978-1973727767



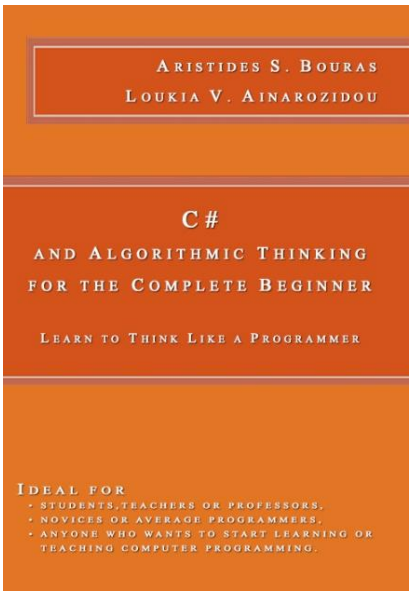
ISBN-10: 1982083670
ISBN-13: 978-1982083670



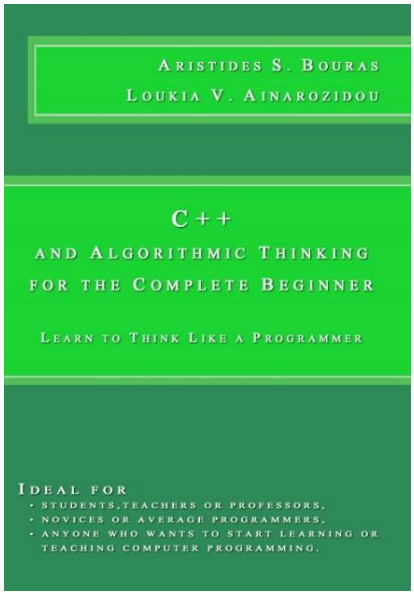
ISBN-10: 1982083697
ISBN-13: 978-1982083694



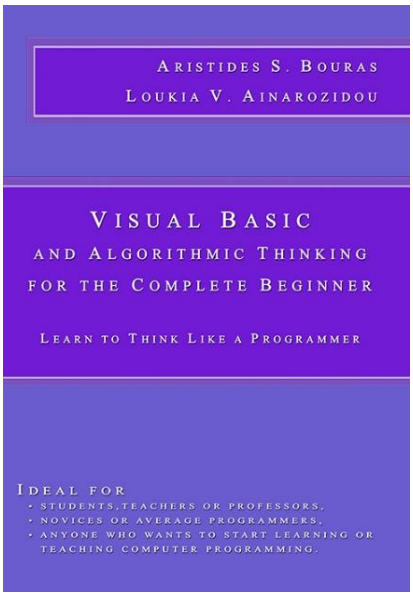
ISBN-10: 1099184878
ISBN-13: 978-1099184871



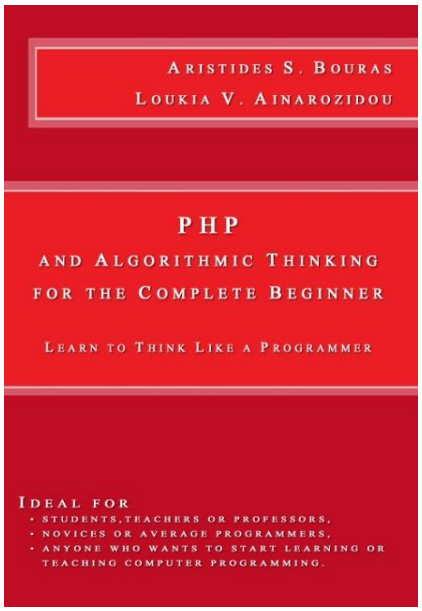
ISBN-10: 1508952485
ISBN-13: 978-1508952480



ISBN-10: 1508577552
ISBN-13: 978-1508577553



ISBN-10: 1511798963
ISBN-13: 978-1511798969



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