PHP
AND ALGORITHMIC THINKING
FOR THE COMPLETE BEGINNER

The Answers

Aristides S. Bouras
Loukia V. Ainarozidou

http://www.bouraspage.com

RCode: 180927

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 "PHP AND ALGORITHMIC THINKING FOR THE COMPLETE BEGINNER". Every effort has been taken to make this book compatible with all previous releases of PHP, and it is almost certain to be compatible with any future releases of PHP.

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 Answers of Review Questions: True/False 8

1.8 Answers of Review Questions: Multiple Choice 8

Chapter 4 9

4.16 Answers of Review Questions: True/False 9

4.17 Answers of Review Questions: Multiple Choice 9

Chapter 5 10

5.9 Answers of Review Questions: True/False 10

5.10 Answers of Review Questions: Multiple Choice 10

5.11 Answers of Review Exercises 10

Chapter 6 11

6.4 Answers of Review Questions: True/False 11

6.5 Answers of Review Questions: Multiple Choice 11

Chapter 7 12

7.7 Answers of Review Questions: True/False 12

7.8 Answers of Review Questions: Multiple Choice 12

7.9 Answers of Review Exercises 12

Chapter 8 13

8.2 Answers of Review Questions: True/False 13

8.3 Answers of Review Exercises 13

Chapter 9 15

9.5 Answers of Review Exercises 15

Chapter 10 17

10.2 Answers of Review Exercises 17

Chapter 11 22

11.3 Answers of Review Questions: True/False 22

11.4 Answers of Review Questions: Multiple Choice 22

11.5 Answers of Review Exercises 22

Chapter 12 24

12.2 Answers of Review Exercises 24
Chapter 13 27
13.2 Answers of Review Exercises 27
Chapter 14 30
14.5 Answers of Review Questions: True/False 30
14.6 Answers of Review Questions: Multiple Choice 30
14.7 Answers of Review Exercises 30
Chapter 15 32
15.8 Answers of Review Questions: True/False 32
15.9 Answers of Review Questions: Multiple Choice 32
15.10 Answers of Review Exercises 32
Chapter 16 35
16.2 Answers of Review Questions: True/False 35
16.3 Answers of Review Questions: Multiple Choice 35
16.4 Answers of Review Exercises 35
Chapter 17 42
17.2 Answers of Review Questions: True/False 42
17.3 Answers of Review Questions: Multiple Choice 42
17.4 Answers of Review Exercises 42
Chapter 18 50
18.2 Answers of Review Questions: True/False 50
18.3 Answers of Review Exercises 50
Chapter 19 57
19.2 Answers of Review Questions: True/False 57
19.3 Answers of Review Exercises 57
Chapter 20 67
20.3 Answers of Review Questions: True/False 67
20.4 Answers of Review Exercises 67
Chapter 21 72
21.13 Answers of Review Questions: True/False 72
21.14 Answers of Review Questions: Multiple Choice 72
21.15 Answers of Review Exercises 72
Chapter 22 81

22.4 Answers of Review Exercises 81

Chapter 23 87

23.6 Answers of Review Exercises 87

Chapter 24 102

24.3 Answers of Review Questions: True/False 102

Chapter 25 103

25.2 Answers of Review Questions: True/False 103

25.3 Answers of Review Questions: Multiple Choice 103

25.4 Answers of Review Exercises 103

Chapter 26 109

26.2 Answers of Review Questions: True/False 109

26.3 Answers of Review Questions: Multiple Choice 109

26.4 Answers of Review Exercises 109

Chapter 27 116

27.3 Answers of Review Questions: True/False 116

27.4 Answers of Review Questions: Multiple Choice 116

27.5 Answers of Review Exercises 116

Chapter 28 127

28.3 Answers of Review Questions: True/False 127

28.4 Answers of Review Questions: Multiple Choice 127

28.5 Answers of Review Exercises 127

Chapter 29 136

29.14 Answers of Review Questions: True/False 136

29.15 Answers of Review Questions: Multiple Choice 136

29.16 Answers of Review Exercises 136

Chapter 30 142

30.4 Answers of Review Exercises 142

Chapter 31 151

31.7 Answers of Review Questions: True/False 151

31.8 Answers of Review Exercises 151

Chapter 32 175

32.3 Answers of Review Questions: True/False 175

32.4 Answers of Review Exercises 175

Chapter 33 178
33.5 Answers of Review Questions: True/False 178
33.6 Answers of Review Questions: Multiple Choice 178
33.7 Answers of Review Exercises 178
Chapter 34 187
34.7 Answers of Review Questions: True/False 187
34.8 Answers of Review Questions: Multiple Choice 187
34.9 Answers of Review Exercises 187
Chapter 35 201
35.7 Answers of Review Questions: True/False 201
35.8 Answers of Review Questions: Multiple Choice 201
35.9 Answers of Review Exercises 201
Chapter 36 212
36.4 Answers of Review Exercises 212
Chapter 37 218
37.7 Answers of Review Questions: True/False 218
37.8 Answers of Review Exercises 219
Chapter 38 249
38.4 Review Questions: True/False 249
Chapter 39 250
39.5 Review Questions: True/False 250
39.6 Review Exercises 250
Chapter 40 257
40.5 Review Questions: True/False 257
40. 6 Review Exercises 257
Chapter 41 264
41.9 Review Questions: True/False 264
41.10 Review Exercises 264
Chapter 42 274
42.4 Review Exercises 274
Chapter 43 280
43.3 Review Exercises 280

How to Report Errata

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

http://www.bouraspage.com

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

Chapter 1

1.7 Answers of Review Questions: True/False

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

1.8 Answers of Review Questions: Multiple Choice

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

Chapter 4

4.16 Answers of Review Questions: True/False

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

4.17 Answers of Review Questions: Multiple Choice

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

Chapter 5

5.9 Answers of Review Questions: True/False

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

5.10 Answers of Review Questions: Multiple Choice

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

5.11 Answers of Review Exercises

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

Chapter 6

6.4 Answers of Review Questions: True/False

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

6.5 Answers of Review Questions: Multiple Choice

1. a
2. b
3. b

Chapter 7

7.7 Answers of Review Questions: True/False

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

7.8 Answers of Review Questions: Multiple Choice

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

7.9 Answers of Review Exercises

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

Chapter 8

8.2 Answers of Review Questions: True/False

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

8.3 Answers of Review Exercises

1. Solution

For the input value of 3

Step	Statement	$a	$b	$c	$d
1	`$a = trim(fgets(STDIN))`	**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	`echo $a, ", ", $b, ", ", $c, ", ", $d`	40 1 1 40 is displayed			

For the input value of 4

Step	Statement	$a	$b	$c	$d
1	`$a = trim(fgets(STDIN))`	**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	`echo $a, ", ", $b, ", ", $c, ", ", $d`	49 10 3 1470 is displayed			

For the input value of 1

Step	Statement	$a	$b	$c	$d
1	`$a = trim(fgets(STDIN))`	**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	`echo $a, ", ", $b, ", ", $c, ", ", $d`	28 2 2 112 is displayed			

2. *Solution*

For the input values of 3, 4

Step	Statement	$a	$b	$c	$d	$e
1	`$a = trim(fgets(STDIN))`	**3**	?	?	?	?
2	`$b = trim(fgets(STDIN))`	3	**4**	?	?	?
3	`$c = $a + $b`	3	4	**7**	?	?
4	`$d = 1 + $a / $b * $c + 2`	3	4	7	**8.25**	?
5	`$e = $c + $d`	3	4	7	8.25	**15.25**
6	`$c += $d + $e`	3	4	**30.5**	8.25	15.25
7	`$e--`	3	4	30.5	8.25	**14.25**
8	`$d -= $c + $d % $c`	3	4	30.5	**-30.25**	14.25
9	`echo $c, ", ",$d, ", ",$e`	30.5 -30.25 14.25 is displayed				

For the input values of 4, 4

Step	Statement	$a	$b	$c	$d	$e
1	`$a = trim(fgets(STDIN))`	**4**	?	?	?	?
2	`$b = trim(fgets(STDIN))`	4	**4**	?	?	?
3	`$c = $a + $b`	4	4	**8**	?	?
4	`$d = 1 + $a / $b * $c + 2`	4	4	8	**11**	?
5	`$e = $c + $d`	4	4	8	11	**19**
6	`$c += $d + $e`	4	4	**38**	11	19
7	`$e--`	4	4	38	11	**18**
8	`$d -= $c + $d % $c`	4	4	38	**-38**	18
9	`echo $c, ", ",$d, ", ",$e`	38 -38 18 is displayed				

Chapter 9

9.5 Answers of Review Exercises

1. Solution

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

2. Solution

For the input values of 5, 5

Step	Statement	$a	$b	$c	$d	$e
1	`$a = trim(fgets(STDIN))`	**5**	?	?	?	?
2	`$b = trim(fgets(STDIN))`	5	**5**	?	?	?
3	`$c = $a + $b`	5	5	**10**	?	?
4	`$d = 5 + $a / $b * $c + 2`	5	5	10	**17**	?
5	`$e = $c - $d`	5	5	10	17	**-7**
6	`$c -= $d + $c`	5	5	**-17**	17	-7
7	`$e--`	5	5	-17	17	**-8**
8	`$d -= $c + $a % $c`	5	5	-17	**29**	-8
9	`echo $c, ", ",$d, ", ",$e`	-17 29 -8 is displayed				

For the input values of 4, 8

Step	Statement	$a	$b	$c	$d	$e
1	`$a = trim(fgets(STDIN))`	**4**	?	?	?	?
2	`$b = trim(fgets(STDIN))`	4	**8**	?	?	?
3	`$c = $a + $b`	4	8	**12**	?	?
4	`$d = 5 + $a / $b * $c + 2`	4	8	12	**13**	?
5	`$e = $c - $d`	4	8	12	13	**-1**
6	`$c -= $d + $c`	4	8	**-13**	13	-1
7	`$e--`	4	8	-13	13	**-2**
8	`$d -= $c + $a % $c`	4	8	-13	**22**	-2
9	`echo $c, ", ",$d, ", ",$e`	-13 22 -2 is displayed				

3. Solution

For the input value of 0.50

Step	Statement	$a	$b	$c
1	`$b = trim(fgets(STDIN))`	?	**0.50**	?
2	`$c = 5`	?	0.50	**5**

3	`$c = $c * $b`	?	0.50	**2.5**
4	`$a = 10 * $c % 10`	**5**	0.50	2.5
5	`echo $a`	Value 5 is displayed		

For the input value of 3

Step	Statement	$a	$b	$c
1	`$b = trim(fgets(STDIN))`	?	3	?
2	`$c = 5`	?	3	**5**
3	`$c = $c * $b`	?	3	**15**
4	`$a = 10 * $c % 10`	**0**	3	15
5	`echo $a`	Value 0 is displayed		

For the input value of 15

Step	Statement	$a	$b	$c
1	`$b = trim(fgets(STDIN))`	?	15	?
2	`$c = 5`	?	15	**5**
3	`$c = $c * $b`	?	15	**75**
4	`$a = 10 * $c % 10`	**0**	15	75
5	`echo $a`	Value 0 is displayed		

Chapter 10

10.2 Answers of Review Exercises

1. Solution

```php
<?php
  echo "Enter base: ";
  $base = trim(fgets(STDIN));
  echo "Enter height: ";
  $height = trim(fgets(STDIN));

  $area = 0.5 * $base * $height;

  echo $area;
?>
```

2. Solution

```php
<?php
  echo "Enter 1st angle: ";
  $angle1 = trim(fgets(STDIN));
  echo "Enter 2nd angle: ";
  $angle2 = trim(fgets(STDIN));

  $angle3 = 180 - $angle1 - $angle2;

  echo $angle3;
?>
```

3. Solution

```php
<?php
  echo "Enter 1st grade: ";
  $g1 = trim(fgets(STDIN));
  echo "Enter 2nd grade: ";
  $g2 = trim(fgets(STDIN));
  echo "Enter 3rd grade: ";
  $g3 = trim(fgets(STDIN));
  echo "Enter 4th grade: ";
  $g4 = trim(fgets(STDIN));

  $average = ($g1 + $g2 + $g3 + $g4) / 4;

  echo $average;
?>
```

4. Solution

```
<?php
  define("PI", 3.14159);
  echo "Enter radius: ";
  $r = trim(fgets(STDIN));

  $perimeter = 2 * PI * $r;

  echo $perimeter;
?>
```

5. Solution

```
<?php
  echo "Enter charge for a meal: ";
  $charge = trim(fgets(STDIN));

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

  $total = $charge + $tip + $tax;

  echo $total;
?>
```

6. Solution

```
<?php
  echo "Enter acceleration in m/sec2: ";
  $a = trim(fgets(STDIN));
  echo "Enter time traveled in sec: ";
  $t = trim(fgets(STDIN));

  $s = 0.5 * $a * $t * $t;

  echo $s;
?>
```

7. Solution

```
<?php
  echo "Enter temperature in Fahrenheit: ";
  $f = trim(fgets(STDIN));

  $c = 5 / 9 * ($f - 32);
  echo $c;
?>
```

8. Solution

```
<?php
  echo "Enter weight in pounds: ";
  $w = trim(fgets(STDIN));
  echo "Enter height in inches: ";
  $h = trim(fgets(STDIN));

  $bmi = $w * 703 / ($h * $h);

  echo $bmi;
?>
```

9. Solution

```
<?php
  echo "Enter subtotal: ";
  $s_total = trim(fgets(STDIN));
  echo "Enter gratuity rate: ";
  $g_rate = trim(fgets(STDIN));

  $tip = $s_total * $g_rate / 100;

  $total = $s_total + $tip;

  echo "Tip is $", $tip;
  echo " and Total is $", $total;
?>
```

10. Solution

```
<?php
  define("VAT", 0.20);

  echo "Enter before-tax price 1: ";
  $btax_price1 = trim(fgets(STDIN));
  echo "Enter before-tax price 2: ";
  $btax_price2 = trim(fgets(STDIN));
  echo "Enter before-tax price 3: ";
  $btax_price3 = trim(fgets(STDIN));

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

```php

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

  echo $avg;
?>
```

11. Solution

```php
<?php
  define("VAT", 0.20);

  echo "Enter after-tax price: ";
  $atax_price = trim(fgets(STDIN));

  $btax_price = $atax_price / (1 + VAT);

  echo $btax_price;
?>
```

12. Solution

```php
<?php
  echo "Enter price: ";
  $i_price = trim(fgets(STDIN));
  echo "Enter discount: ";
  $discount = trim(fgets(STDIN));

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

  echo $f_price, " ", $saved;
?>
```

13. Solution

```php
<?php
  define("VAT", 0.20);

  echo "Enter kWh at the beginning of the month: ";
  $i_kWh = trim(fgets(STDIN));
  echo "Enter kWh at the end of the month: ";
  $f_kWh = trim(fgets(STDIN));

  $kWh_consumed = $f_kWh - $i_kWh;

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

  echo $kWh_consumed, " ", $cost;
?>
```

14. Solution

```
<?php
  echo "Enter day: ";
  $day = trim(fgets(STDIN));
  echo "Enter month: ";
  $month = trim(fgets(STDIN));

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

  echo $days_left;
?>
```

Chapter 11

11.3 Answers of Review Questions: True/False

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

11.4 Answers of Review Questions: Multiple Choice

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

11.5 Answers of Review Exercises

1. Solution

For the input value of 9

Step	Statement	$a	$b	$c
1	`$a = trim(fgets(STDIN))`	**9**	?	?
2	`$a += 6 / sqrt($a) * 2 + 20`	**33**	?	**?**
3	`$b = round($a, 0) % 4`	33	**1**	**?**
4	`$c = $b % 3`	33	1	**1**
5	`echo $a, ", ", $b, ", ",$c`	33 1 1 is displayed		

For the input value of 4

Step	Statement	$a	$b	$c
1	`$a = trim(fgets(STDIN))`	**4**	?	?
2	`$a += 6 / sqrt($a) * 2 + 20`	**30**	?	**?**
3	`$b = round($a, 0) % 4`	30	**2**	**?**
4	`$c = $b % 3`	30	2	**2**
5	`echo $a, ", ", $b, ", ",$c`	30 2 2 is displayed		

2. Solution

For the input value of -2

Step	Statement	$a	$b	$c
1	`$a = trim(fgets(STDIN))`	**-2**	?	?
2	`$b = abs($a) % 4 + pow($a , 4)`	-2	**18**	**?**
3	`$c = $b % 5`	-2	18	**3**

4	`echo $b, ", ",$c`	18, 3 is displayed

For the input value of -3

Step	Statement	$a	$b	$c
1	`$a = trim(fgets(STDIN))`	**-3**	?	?
2	`$b = abs($a) % 4 + pow($a , 4)`	-3	**84**	**?**
3	`$c = $b % 5`	-3	84	**4**
4	`echo $b, ", ",$c`	84 4 is displayed		

3. Solution

```
<?php
  echo "Enter angle in radians: ";
  $radians = trim(fgets(STDIN));

  $degrees = $radians * 180 / pi();

  echo $degrees;
?>
```

4. Solution

```
<?php
  echo "Enter side A of a right-angled triangle: ";
  $a = trim(fgets(STDIN));
  echo "Enter side B of a right-angled triangle: ";
  $b = trim(fgets(STDIN));

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

  echo $hypotenuse;
?>
```

5. Solution

```
<?php
  echo "Enter angle (in degrees) of a right-angled triangle: ";
  $th = trim(fgets(STDIN));
  echo "Enter length of adjacent side: ";
  $adjacent = trim(fgets(STDIN));

  $opposite = tan($th * pi() / 180) * $adjacent;

  echo $opposite;
?>
```

Chapter 12

12.2 Answers of Review Exercises

1. *Solution*

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

2. *Solution*

i. `$y = pow($x + 3, 5 * $w) / (7 * ($x - 4))`
ii. `$y = pow(3 * pow($x, 2) – pow($x, 3) / 4, 1 / 5)`
iii. `$y = sqrt(pow($x, 4) – 2 * pow($x, 3) – 7 * $x * $x +$x) / pow(4 * (7 * pow($x, 4) – 3 / 4 * pow($x, 3)) * (7 * $x * $x + x), 1/3)`
iv. `$y = $x / ($x - 3 * ($x - 1)) + $x * pow($x - 1, 1 / 5) / ((pow($x, 3) - 2) * pow($x - 1, 3))`
v. `$y = pow(sin(pi() / 3) – cos(pi() / 2 * $w) , 2)`
vi. `$y = pow(sin(pi() / 2 * $x) + cos(3 * pi() / 2 * $w), 3) / pow(tan(2 * pi() / 3 * $w) - sin(pi() / 2 * $x), 0.5)`

3. *Solution*

```
<?php
  echo "Enter value for x: ";
  $x = trim(fgets(STDIN));

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

  echo $y;
?>
```

4. *Solution*

```
<?php
  echo "Enter value for x: ";
  $x = trim(fgets(STDIN));

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

  echo $y;
?>
```

5. *Solution*

```
<?php
  echo "Enter value for x: ";
  $x = trim(fgets(STDIN));
  echo "Enter value for w: ";
  $w = trim(fgets(STDIN));

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

  echo $y;
?>
```

6. Solution

```
<?php
  echo "Enter value for x: ";
  $x = trim(fgets(STDIN));
  echo "Enter value for w: ";
  $w = trim(fgets(STDIN));

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

  echo $y;
?>
```

7. Solution

```
<?php
  echo "Enter value for x: ";
  $x = trim(fgets(STDIN));
  echo "Enter value for w: ";
  $w = trim(fgets(STDIN));

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

  echo $y;
?>
```

8. Solution

```
<?php
  echo "Enter length A: ";
  $a = trim(fgets(STDIN));
  echo "Enter length B: ";
  $b = trim(fgets(STDIN));
  echo "Enter length C: ";
  $c = trim(fgets(STDIN));

  $semi = ($a + $b + $c) / 2;
  $area = sqrt($semi * ($semi - $a) * ($semi - $b) * ($semi - $c));
  echo $area;
?>
```

Chapter 13

13.2 Answers of Review Exercises

1. *Solution*

```
<?php
  echo "Enter an integer: ";
  $n = trim(fgets(STDIN));

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

  echo $result;
?>
```

2. *Solution*

```
<?php
  echo "Enter a five-digit integer: ";
  $number = trim(fgets(STDIN));

  $digit5 = $number % 10;
  $r = intval($number / 10);

  $digit4 = $r % 10;
  $r = intval($r / 10);

  $digit3 = $r % 10;
  $r = intval($r / 10);

  $digit2 = $r % 10;
  $digit1 = intval($r / 10);

  $reversed = $digit5 * 10000 + $digit4 * 1000 + $digit3 * 100 + $digit2 * 10 + $digit1;
  echo $reversed;
?>
```

3. *Solution*

```
<?php
  echo "Enter an integer: ";
  $n = trim(fgets(STDIN));

  $result = $n % 2;

  echo $result;
```

```
?>
```

4. Solution

```
<?php
  echo "Enter an integer: ";
  $n = trim(fgets(STDIN));

  $result = 1 - $n % 2;

  echo $result;
?>
```

5. Solution

```
<?php
  echo "Enter a period of time in seconds: ";
  $number = trim(fgets(STDIN));

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

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

  $hours = intval($r / 3600);
  $r = $r % 3600;

  $minutes = intval($r / 60);
  $seconds = $r % 60;

  echo $weeks, " weeks ", $days, " days ", $hours, " hours ";
  echo $minutes, " minutes and ", $seconds, " seconds";
?>
```

6. Solution

```
<?php
  echo "Enter amount to withdraw: ";
  $amount = trim(fgets(STDIN));

  $usd20 = intval($amount / 20);
  $r = $amount % 20;

  $usd10 = intval($r / 10);
  $r = $r % 10;

  $usd5 = intval($r / 5);
  $usd1 = $r % 5;

  echo $usd20, " notes of $20 ", $usd10, " notes of $10 ";
  echo $usd5, " notes of $5 and ", $usd1, " notes of $1";
?>
```

7. Solution

```
<?php
  echo "Enter number of steps: ";
  $steps = trim(fgets(STDIN));

  $distance = $steps * 25;

  $miles = intval($distance / 63360);
  $r = $distance % 63360;

  $yards = intval($r / 36);
  $r = $r % 36;

  $feet = intval($r / 12);
  $inches = $r % 12;

  echo $miles, " miles ", $yards, " yards ";
  echo $feet, " feet and ", $inches, " inches";
?>
```

Chapter 14

14.5 Answers of Review Questions: True/False

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

14.6 Answers of Review Questions: Multiple Choice

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

14.7 Answers of Review Exercises

1. Solution

```php
<?php
  echo "First name: ";
  $first_name = trim(fgets(STDIN));
  echo "Middle name: ";
  $middle_name = trim(fgets(STDIN));
  echo "Last name: ";
  $last_name = trim(fgets(STDIN));
  echo "Title: ";
  $title = trim(fgets(STDIN));

  echo $title, " ", $first_name, " ", $middle_name, " ", $last_name, "\n";
  echo $first_name, " ", $middle_name, " ", $last_name, "\n";
  echo $last_name, ", ", $first_name, "\n";
  echo $last_name, ", ", $first_name, " ", $middle_name, "\n";
  echo $last_name, ", ", $first_name, " ", $middle_name, ", ", $title, "\n";
  echo $first_name, " ", $last_name, "\n";
?>
```

2. Solution

```php
<?php
  $alphabet = "abcdefghijklmnopqrstuvwxyz";

  $rnd_word = strtoupper($alphabet[rand(0, 25)]);
  $rnd_word .= $alphabet[rand(0, 25)] . $alphabet[rand(0, 25)] .
               $alphabet[rand(0, 25)] . $alphabet[rand(0, 25)];

  echo $rnd_word;
?>
```

3. Solution

```
<?php
  echo "Enter name: ";
  $name = trim(fgets(STDIN));

  $name = strtolower($name);

  $password = "";
  $password .= $name[rand(0, strlen($name) - 1)];
  $password .= $name[rand(0, strlen($name) - 1)];
  $password .= $name[rand(0, strlen($name) - 1)];
  $password .= rand(1000, 9999);

  echo $password;
?>
```

Chapter 15

15.8 Answers of Review Questions: True/False

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

15.9 Answers of Review Questions: Multiple Choice

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

15.10 Answers of Review Exercises

1. Solution

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

2. Solution

$a	$b	$c	$a != 1	$b > $a	$c / 2 > 2 * $a
3	**-5**	**8**	True	False	False
1	**10**	**20**	False	True	True
-4	**-2**	**-9**	True	True	True

3. Solution

Boolean Expression1 (BE1)	Boolean Expression2 (BE2)	BE1 || BE2	BE1 && BE2	!(BE2)
False	**False**	False	False	True
False	**True**	True	False	False

True	**False**	True	False	True
True	**True**	True	True	False

4. Solution

$a	$b	$c	$a > 3 || $c > $b && $c > 1	$a > 3 && $c > $b || $c > 1
4	**-6**	**2**	True	True
-3	**2**	**-4**	False	False
2	**5**	**5**	False	True

5. Solution

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

6. Solution

i. false

ii. true

7. Solution

i. `$age < 12 && $age != 8`

ii. `$age >= 6 && $age <= 9 || $age == 11`

iii. `$age > 7 && $age != 10 && $age != 12`

iv. `$age == 6 || $age == 9 || $age == 11`

v. `$age >= 6 && $age <= 12 && $age != 8`

vi. `$age != 7 && $age != 10`

8. Solution

i. `$x != 4 || $y == 3`

ii. `$x + 4 > 0`

iii. `!($x <= 5) && $y != 4`

iv. `$x == false`

v. `!($x < 4 && $z <= 4)`

vi. `$x == 2 || $x < -5`

9. Solution

i. `!($x < 4 || $y == 10)`

ii. `!( $x - 2 < 9)`

iii. `!( !($x < 2) && $y == 4)`

iv. `!( $x == false && $y != 3)`

v. First approach: `!(!($x < 2 || $y < 2))`

Second approach: `$x < 2 || $y < 2`

vi. `!( $x == -2 || $x > 2)`

Chapter 16

16.2 Answers of Review Questions: True/False

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

16.3 Answers of Review Questions: Multiple Choice

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

16.4 Answers of Review Exercises

1. Solution

The corrections/additions are in red

```
<?php
  $x = trim(fgets(STDIN));

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

  echo $x, $y;
?>
```

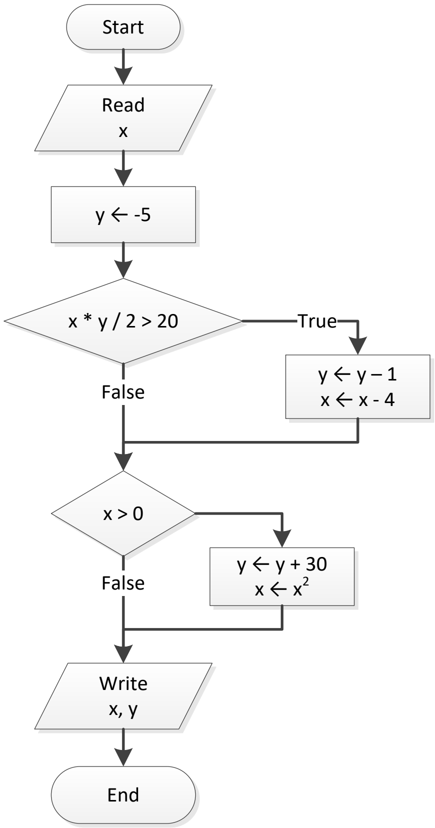
2. Solution

For the input value of 10

Step	Statement	$x	$y
1	`$x = trim(fgets(STDIN))`	**10**	?
2	`$y = - 5`	10	**-5**
3	`if ($x * $y / 2 > 20)`	False	
4	`if ($x > 0)`	True	
5	`$y += 30`	10	**25**
6	`$x = pow($x,2)`	**100**	25
7	`echo $x, $y`	100 25 is displayed	

For the input value of -10

Step	Statement	$x	$y
1	$x = trim(fgets(STDIN))	**-10**	?
2	$y = - 5	-10	**-5**
3	if ($x * $y / 2 > 20)	True	
4	$y--	-10	**-6**
5	$x -= 4	**-14**	-6
6	if ($x > 0)	False	
7	echo $x, $y	-14 -6 is displayed	



3. *Solution*

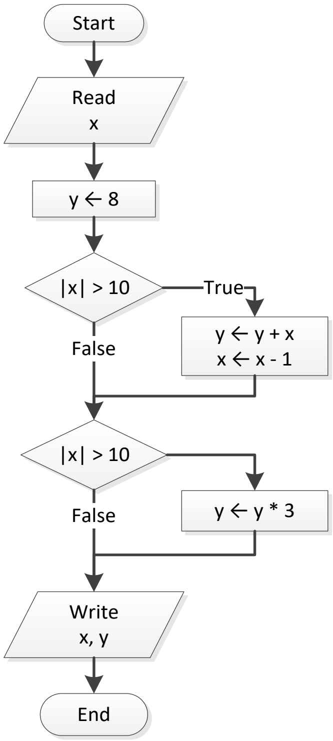
For the input value of -11

Step	Statement	$x	$y
1	$x = trim(fgets(STDIN))	**-11**	?
2	$y = 8	-11	**8**
3	if (abs($x) > 10)	True	
4	$y += $x	-11	**-3**
5	$x--	**-12**	-3

6	`if (abs($x) > 10)`	True	
7	`$y *= 3`	-12	**-9**
8	`echo $x, $y`	-12 -9 is displayed	

For the input value of 11

Step	Statement	$x	$y
1	`$x = trim(fgets(STDIN))`	**11**	?
2	`$y = 8`	11	**8**
3	`if (abs($x) > 10)`	True	
4	`$y += $x`	11	**19**
5	`$x--`	**10**	19
6	`if (abs($x) > 10)`	False	
7	`echo $x, $y`	10 19 is displayed	



4. Solution

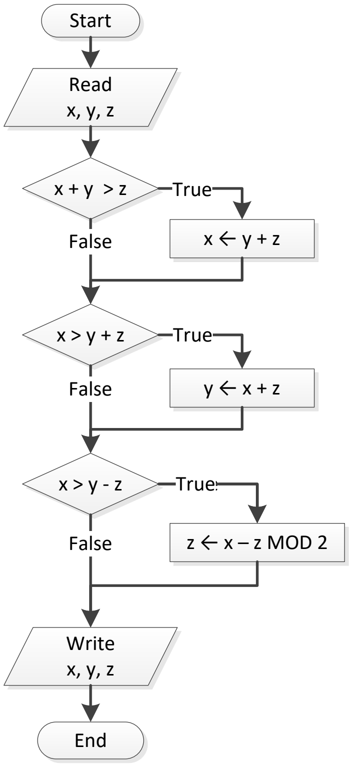
For input values of 1, 2 and 3

Step	Statement	$x	$y	$z
1	`$x = trim(fgets(STDIN))`	**1**	?	?
2	`$y = trim(fgets(STDIN))`	1	**2**	?
3	`$z = trim(fgets(STDIN))`	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	`echo $x, $y, $z`	1 2 0 is displayed		

For input values of 4, 2 and 1

Step	Statement	$x	$y	$z
1	`$x = trim(fgets(STDIN))`	**4**	?	?
2	`$y = trim(fgets(STDIN))`	4	**2**	?
3	`$z = trim(fgets(STDIN))`	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	`echo $x, $y, $z`	3 2 2 is displayed		



5. Solution

```
<?php
  echo "Enter a number: ";
  $x = trim(fgets(STDIN));

  if ($x > 0) {
    echo "Positive";
  }
?>
```

6. Solution

```
<?php
  echo "Enter a number: ";
  $x = trim(fgets(STDIN));
  echo "Enter a second number";
  $y = trim(fgets(STDIN));

  if ($x > 0 && $y > 0) {
    echo "Positive";
  }
?>
```

7. Solution

```
<?php
  echo "Enter a number: ";
  $x = trim(fgets(STDIN));

  if (is_numeric($x) == true) {
    echo "Numeric";
  }
?>
```

8. Solution

```
<?php
  echo "Enter a string: ";
  $str = trim(fgets(STDIN));

  if ($str == strtoupper($str)) {
    echo "Uppercase";
  }
?>
```

9. Solution

```
<?php
  echo "Enter a string: ";
  $str = trim(fgets(STDIN));

  if (strlen($str) > 20) {
    echo "Many characters";
  }
?>
```

10. Solution

```
<?php
  echo "Enter 1st number: ";
  $n1 = trim(fgets(STDIN));
  echo "Enter 2nd number: ";
  $n2 = trim(fgets(STDIN));
  echo "Enter 3rd number: ";
  $n3 = trim(fgets(STDIN));
  echo "Enter 4th number: ";
  $n4 = trim(fgets(STDIN));

  if ($n1 < 0 || $n2 < 0 || $n3 < 0 || $n4 < 0) {
    echo "Among the given numbers, there is a negative one!";
  }
?>
```

11. Solution

```
<?php
  echo "Enter 1st number: ";
  $a = trim(fgets(STDIN));
  echo "Enter 2nd number: ";
  $b = trim(fgets(STDIN));

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

  echo $a, $b;
?>
```

12. Solution

```
<?php
  echo "Enter 1st temperature: ";
  $t1 = trim(fgets(STDIN));
  echo "Enter 2nd temperature: ";
  $t2 = trim(fgets(STDIN));
  echo "Enter 3rd temperature: ";
  $t3 = trim(fgets(STDIN));
  $average = ($t1 + $t2 + $t3) / 3;

  if ($average > 60) {
    echo "Heat Wave";
  }
?>
```

Chapter 17

17.2 Answers of Review Questions: True/False

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

17.3 Answers of Review Questions: Multiple Choice

1. b
2. c
3. c

17.4 Answers of Review Exercises

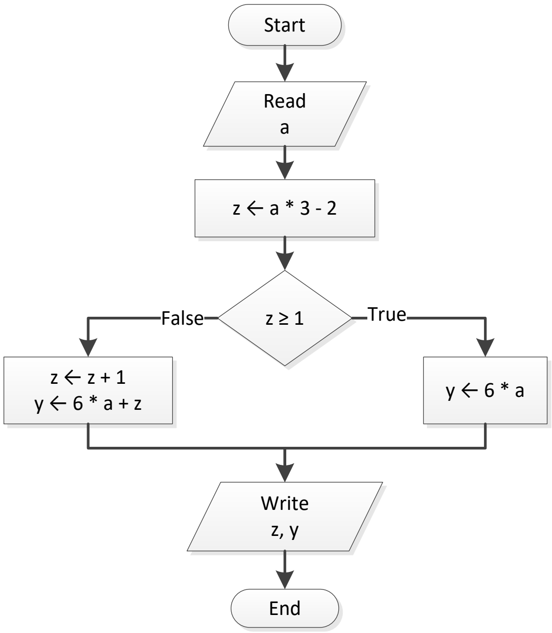
1. Solution

For input value of 3

Step	Statement	$a	$y	$z
1	`$a = trim(fgets(STDIN))`	**3**	?	?
2	`$z = $a * 3 - 2`	3	?	**7**
3	`if ($z >= 1)`	True		
4	`$y = 6 * $a`	3	**18**	7
5	`echo $z, $y`	7 18 is displayed		

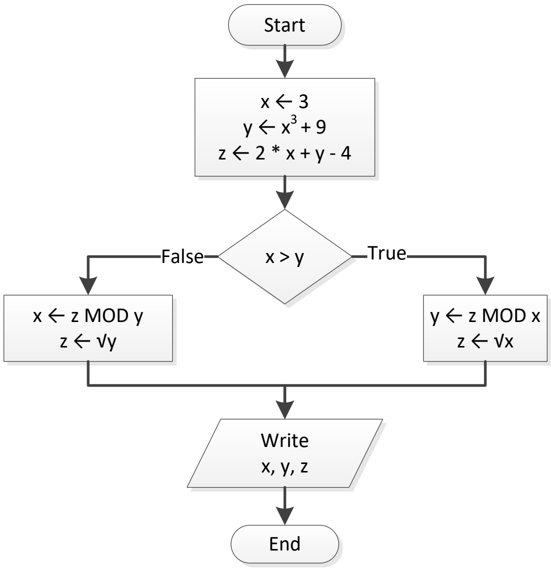
For input value of 0.5

Step	Statement	$a	$y	$z
1	`$a = trim(fgets(STDIN))`	**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	`echo $z, $y`	0.5 3.5 is displayed		



2. Solution

Step	Statement	$x	$y	$z
1	`$x = 3`	**3**	?	?
2	`$y = pow($x, 3) + 9`	3	**36**	?
3	`$z = 2 * $x + $y - 4`	3	36	**38**
4	`if ($x > $y)`	False		
5	`$x = $z % $y`	**2**	36	38
6	`$z = sqrt($y)`	2	36	**6**
7	`echo $x, $y, $z`	2 36 6 is displayed		



3. *Solution*

```
<?php
  $x = trim(fgets(STDIN));
  $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;
  }
  echo $y;
?>
```

For input value of 10

Step	Statement	$x	$y	$w	$z
1	`$x = trim(fgets(STDIN))`	**10**	?	?	?
2	`$w = $x * 3 - 15`	10	?	**15**	?
3	`$z = ($w + 7) * ($x + 4) - 10`	10	?	15	**298**
4	`if ($w > $x && $z > $x)`	True			
5	`$x++`	**11**	?	15	298
6	`$y = $x / 2 + 4`	11	**9.5**	15	298
7	`echo $y`	9.5 is displayed			

For input value of 2

Step	Statement	$x	$y	$w	$z
1	`$x = trim(fgets(STDIN))`	**2**	?	?	?

2	`$w = $x * 3 - 15`	2	?	**-9**	?
3	`$z = ($w + 7) * ($x + 4) - 10`	2	?	-9	**-22**
4	`if ($w > $x && $z > $x)`	False			
5	`$y = $x / 4 + 2`	2	**2.5**	-9	-22
6	`echo $y`	2.5 is displayed			

4. Solution

```
<?php
  $x = trim(fgets(STDIN));
  if ($x % 6 == 0) {
    echo $x, " is a multiple of 6";
  }
  else {
    echo $x, " is not a multiple of 6";
  }
?>
```

5. Solution

```
<?php
  $x = trim(fgets(STDIN));
  if ($x % 6 == 0 || $x % 7 == 0) {
    echo $x, " is a multiple of 6 or a multiple of 7";
  }
  else {
    echo $x, " is neither a multiple of 6 nor a multiple of 7";
  }
?>
```

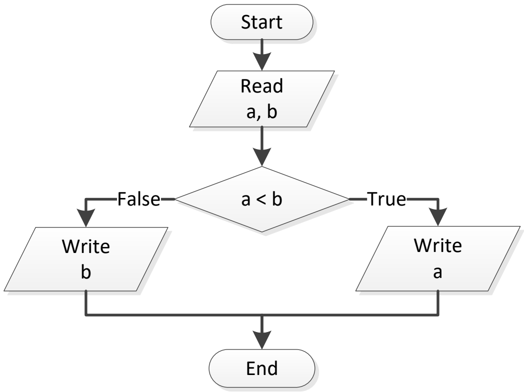
6. Solution

```
<?php
  $x = trim(fgets(STDIN));

  $y = $x % 4;
  if ($y == 0) {
    echo $x, " is a multiple of 4\n";
  }
  else {
    echo $x, " is not a multiple of 4\n";
  }

  echo "The structure is: ", $x, " = ", intval($x / 4), " x 4 + ", $y;
?>
```

7. Solution

```
<?php
  $x = trim(fgets(STDIN));

  if ($x >= 1000 && $x <= 9999) {
    echo $x, " is a four-digit integer";
  }
  else {
    echo $x, " is not a four-digit integer";
  }
?>
```

8. Solution



```
<?php
  $a = trim(fgets(STDIN));
  $b = trim(fgets(STDIN));

  if ($a < $b) {
    echo $a;
  }
  else {
    echo $b;
  }
?>
```

9. Solution

```
<?php
  $a = trim(fgets(STDIN));
  $b = trim(fgets(STDIN));
  $c = trim(fgets(STDIN));

  if ($a < $b + $c && $b < $a + $c && $c < $a + $b) {
    echo "Given numbers can be lengths of the three sides of a triangle";
  }
  else {
    echo "Given numbers cannot be lengths of the three sides of a triangle";
  }
?>
```

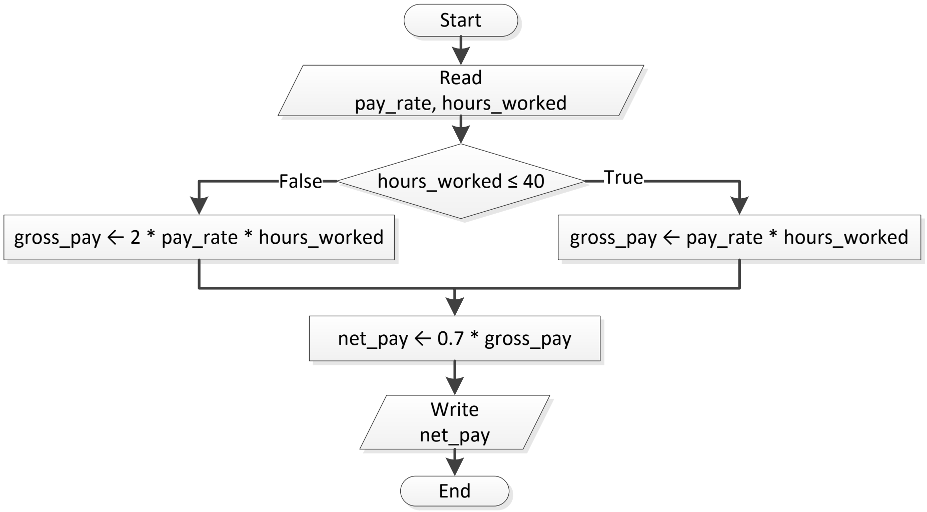
10. Solution

```
<?php
  $a = trim(fgets(STDIN));
  $b = trim(fgets(STDIN));
  $c = trim(fgets(STDIN));

  if (pow($a, 2) == pow($b, 2) + pow($c, 2) ||
      pow($b, 2) == pow($a, 2) + pow($c, 2) ||
      pow($c, 2) == pow($a, 2) + pow($b, 2)) {
    echo "Given numbers can be lengths of the three sides of a right triangle";
  }
  else {
    echo "Given numbers cannot be lengths of the three sides of a right triangle";
  }
?>
```

11. Solution

```
<?php
  echo "Enter 1st jump in meters: ";
  $a = trim(fgets(STDIN));
  echo "Enter 2nd jump in meters: ";
  $b = trim(fgets(STDIN));
  echo "Enter 3rd jump in meters: ";
  $c = trim(fgets(STDIN));

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

  if ($average < 8) {
    echo "Disqualified";
  }
  else {
    echo "Qualified";
  }
?>
```

12. Solution


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

```
<?php
  $pay_rate = trim(fgets(STDIN));
  $hours_worked = trim(fgets(STDIN));

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

  $net_pay = 0.7 * $gross_pay;
  echo "Net Pay: ", $net_pay;
?>
```

13. Solution

```
<?php
  echo "Enter miles traveled: ";
  $miles = trim(fgets(STDIN));

  $r = $miles % 12000;

  if ($r > 6000) {
    $miles_left = 12000 - $r;
    echo "Your car needs a major service in ", $miles_left, " miles";
  }
  else {
    $miles_left = 6000 - $r;
    echo "Your car needs a minor service in ", $miles_left, " miles";
  }
?>
```

14. Solution

```
<?php
  echo "Enter the time the two cars traveled: ";
  $t = trim(fgets(STDIN));
  echo "Enter the accelaration for car A: ";
  $a1 = trim(fgets(STDIN));
  echo "Enter the accelaration for car B: ";
  $a2 = trim(fgets(STDIN));

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

  if ($s1 > $s2) {
    echo "Car A is first";
  }
  else {
    echo "Car B is first";
  }
?>
```

Chapter 18

18.2 Answers of Review Questions: True/False

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

18.3 Answers of Review Exercises

1. Solution

For input value of 5

Step	Statement	$q	$b
1	`$q = trim(fgets(STDIN))`	**5**	?
2	`if ($q > 0 && $q <= 50)`	True	
3	`$b = 1`	5	**1**
4	`echo $b`	1 is displayed	

For input value of 150

Step	Statement	$q	$b
1	`$q = trim(fgets(STDIN))`	**150**	?
2	`if ($q > 0 && $q <= 50)`	False	
3	`elseif ($q > 50 && $q <= 100)`	False	
4	`elseif ($q > 100 && $q <= 200)`	True	
5	`$b = 3`	150	**3**
6	`echo $b`	3 is displayed	

For input value of 250

Step	Statement	$q	$b
1	`$q = trim(fgets(STDIN))`	**250**	?
2	`if ($q > 0 && $q <= 50)`	False	
3	`elseif ($q > 50 && $q <= 100)`	False	
4	`elseif ($q > 100 && $q <= 200)`	False	
5	`$b = 4`	250	**4**
6	`echo $b`	4 is displayed	

For input value of -1

Step	Statement	$q	$b
1	`$q = trim(fgets(STDIN))`	**-1**	?
2	`if ($q > 0 && $q <= 50)`	False	

3	`elseif ($q > 50 && $q <= 100)`	False	
4	`elseif ($q > 100 && $q <= 200)`	False	
5	`$b = 4`	-1	**4**
6	`echo $b`	4 is displayed	

2. *Solution*

For input value of 5

Step	Statement	$amount	$discount	$payment
1	`$amount = trim(fgets(STDIN))`	**5**	?	?
2	`if ($amount < 20)`	True		
3	`$discount = 0`	5	**0**	?
4	`$payment = $amount - $amount * $discount / 100`	5	0	**5**
5	`echo $discount, $payment`	0 5 is displayed.		

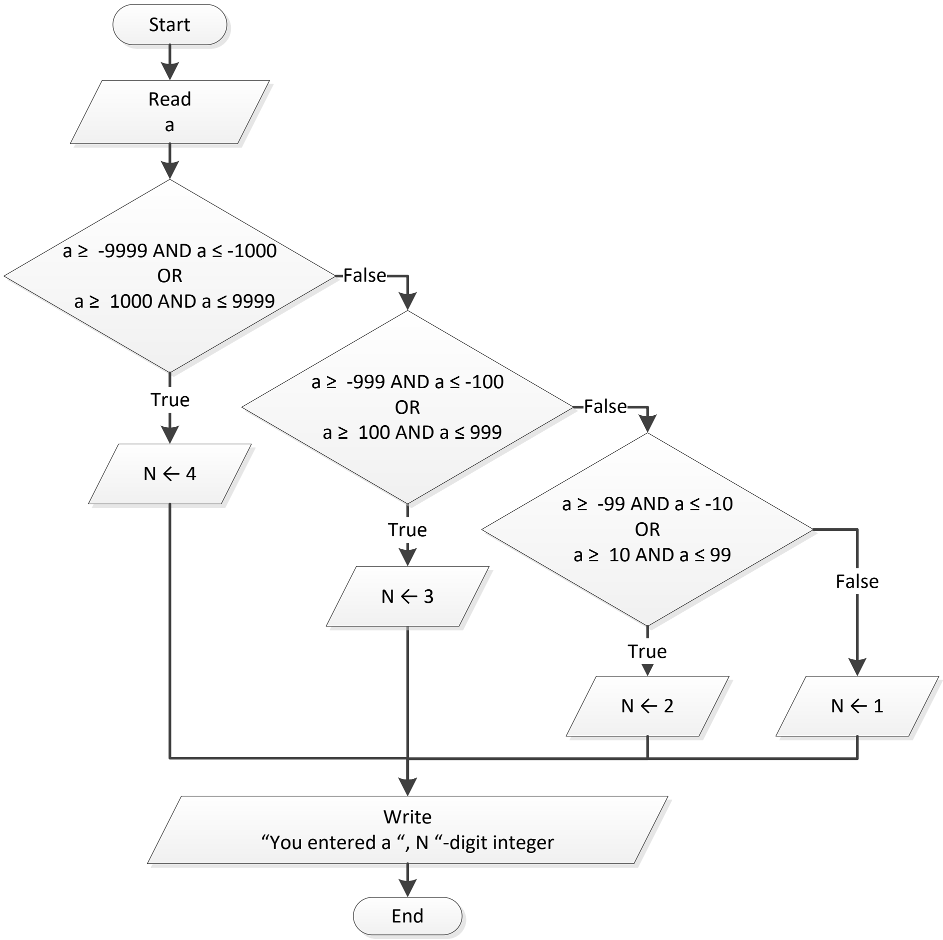
For input value of 150

Step	Statement	$amount	$discount	$payment
1	`$amount = trim(fgets(STDIN))`	**150**	?	?
2	`if ($amount < 20)`	False		
3	`elseif ($amount >=20 && $amount < 60)`	False		
4	`elseif ($amount >= 60 && $amount < 100)`	False		
5	`elseif ($amount >= 100)`	True		
6	`$discount = 15`	150	**15**	?
7	`$payment = $amount - $amount * $discount / 100`	150	15	**5**
8	`echo $discount, $payment`	15 127.5 is displayed.		

For input value of -1

Step	Statement	$amount	$discount	$payment
1	`$amount = trim(fgets(STDIN))`	**-1**	?	?
2	`if ($amount < 20)`	True		
3	`$discount = 0`	-1	**0**	?
4	`$payment = $amount - $amount * $discount / 100`	-1	0	**-1**
5	`echo $discount, $payment`	0 -1 is displayed.		

3. Solution



```
<?php
  $a = trim(fgets(STDIN));

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

  echo "You entered a ", $n, "-digit integer";
?>
```

4. Solution

```
<?php
  echo "1. Convert USD to Euro (EUR)\n";
  echo "2. Convert USD to British Pound Sterling (GBP)\n";
  echo "3. Convert USD to Japanese Yen (JPY)\n";
  echo "4. Convert USD to Canadian Dollar (CAD)\n";

  echo "Enter a choice: ";
  $ch = trim(fgets(STDIN));

  echo "Enter an amount in US dollars: ";
  $usd = trim(fgets(STDIN));

  if ($ch == 1) {
    $eur = $usd / 0.72;
    echo "$", $usd, " = ", $eur, " EUR";
  }
  elseif($ch == 2) {
    $gbp = $usd / 0.60;
    echo "$", $usd, " = ", $gbp, " GBP";
  }
  elseif($ch == 3) {
    $jpy = $usd / 102.15;
    echo "$", $usd, " = ", $jpy, " JPY";
  }
  else {
    $cad = $usd / 1.10;
    echo "$", $usd, " = ", $cad, " CAD";
  }
?>
```

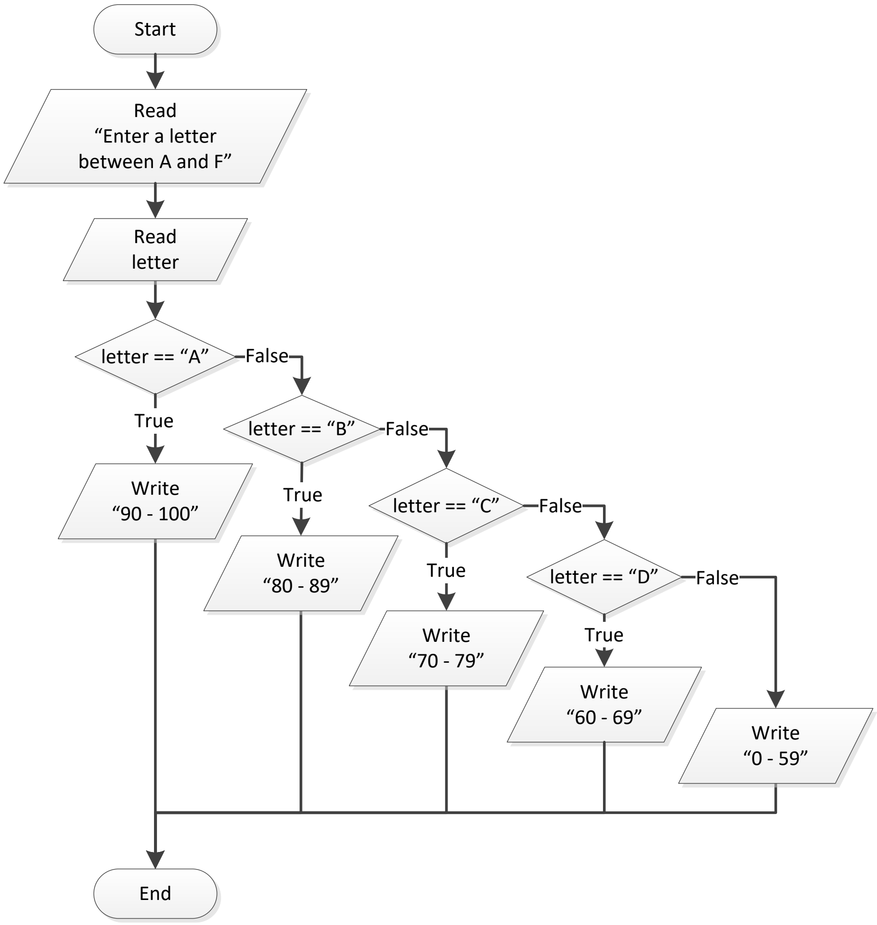
5. Solution

```
<?php
  echo "Enter the number of a month between 1 and 12: ";
  $m = trim(fgets(STDIN));

  if ($m <= 2 || $m == 12) {
    echo "Winter";
  }
  elseif ($m <= 5) {
    echo "Spring";
  }
  elseif ($m <= 8) {
    echo "Summer";
  }
  else {
    echo "Fall (Autumn)";
  }
?>
```

6. *Solution*

```
<?php
  echo "Enter a number between 1.0 and 4.9: ";
  $n = trim(fgets(STDIN));

  $x = intval($n);
  $y = $n * 10 % 10;

  if ($x == 1) {
    echo "One";
  }
  elseif ($x == 2) {
    echo "Two";
  }
  elseif ($x == 3) {
    echo "Three";
  }
  elseif ($x == 4) {
    echo "Four";
  }

  echo " point ";

  if ($y == 1) {
    echo "one";
  }
  elseif ($y == 2) {
    echo "two";
  }
  elseif ($y == 3) {
    echo "three";
  }
  elseif ($y == 4) {
    echo "four";
  }
  elseif ($y == 5) {
    echo "five";
  }
  elseif ($y == 6) {
    echo "six";
  }
  elseif ($y == 7) {
    echo "seven";
  }
  elseif ($y == 8) {
    echo "eight";
  }
  elseif ($y == 9) {
    echo "nine";
  }
  elseif ($y == 0) {
    echo "zero";
  }
?>
```

7. Solution



```
<?php
  echo "Enter a letter between A and F: ";
  $letter = trim(fgets(STDIN));
  if ($letter == "A") {
    echo "90 - 100";
  }
  elseif ($letter == "B") {
    echo "80 - 89";
  }
  elseif( $letter == "C") {
    echo "70 - 79";
  }
  elseif ($letter == "D") {
    echo "60 - 69";
  }
  else {
    echo "0 - 59";
  }
?>
```

Chapter 19

19.2 Answers of Review Questions: True/False

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

19.3 Answers of Review Exercises

1. Solution

For input value of 1

Step	Statement	$a	$x	y
1	`$a = trim(fgets(STDIN))`	**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	`echo $x, $y`	5 5 is displayed		

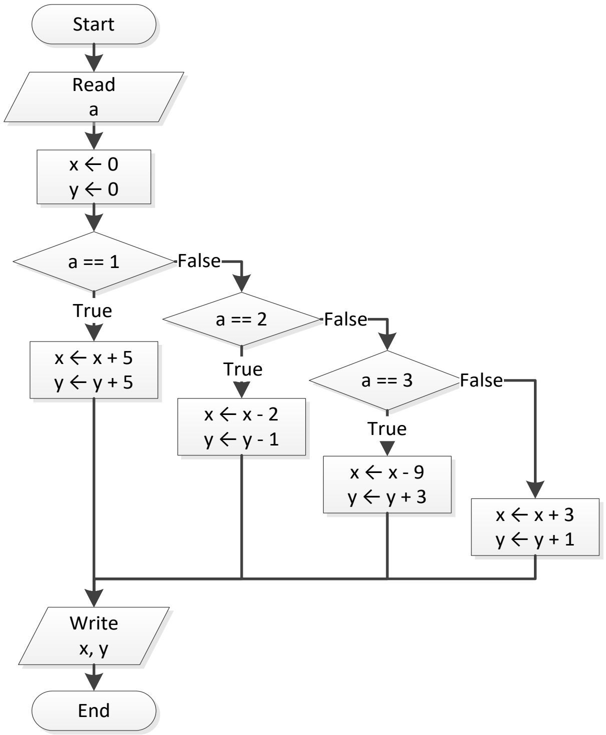
For input value of 3

Step	Statement	$a	$x	y
1	`$a = trim(fgets(STDIN))`	**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	`echo $x, $y`	-9 3 is displayed		

For input value of 250

Step	Statement	$a	$x	y
1	`$a = trim(fgets(STDIN))`	**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	`echo $x, $y`	3 1 is displayed		



2. *Solution*

For input values of 10, 2, 5

Step	Statement	$a	$x	y
1	`$a = trim(fgets(STDIN))`	**10**	?	?
2	`$x = trim(fgets(STDIN))`	10	**2**	?
3	`$y = trim(fgets(STDIN))`	10	2	**5**
4	`case $a == 10`	True		
5	`$x = $x % 2`	10	**0**	5
6	`$y = pow($y, 2)`	10	0	**25**
7	`echo $x, $y`	0 25 is displayed		

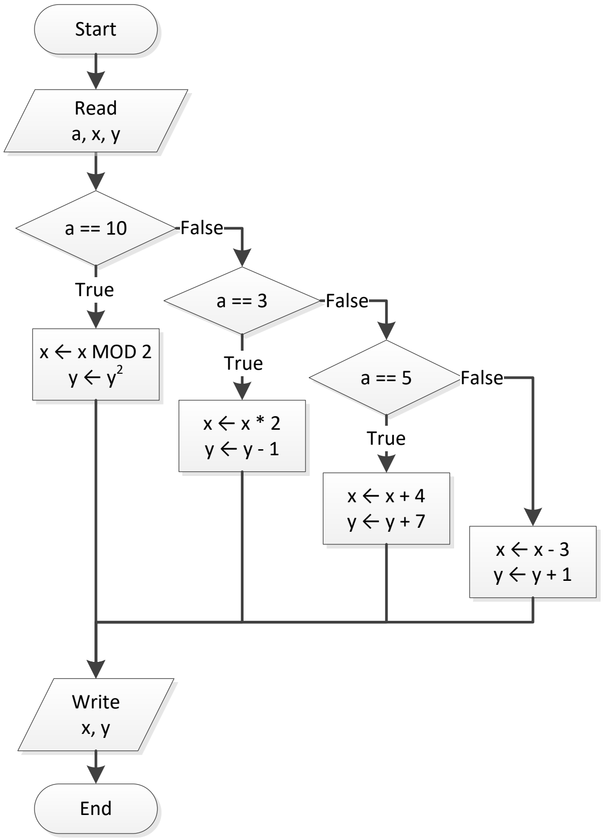
For input values of 5, 2, 3

Step	Statement	$a	$x	y
1	`$a = trim(fgets(STDIN))`	**5**	?	?
2	`$x = trim(fgets(STDIN))`	5	**2**	?

3	`$y = trim(fgets(STDIN))`	5	2	**3**
4	`case $a == 10`	False		
5	`case $a == 3`	False		
6	`case $a == 5`	True		
7	`$x = $x + 4`	5	**6**	3
8	`$y += 7`	5	6	**10**
9	`echo $x, $y`	6 10 is displayed		

For input values of 4, 6, 2

Step	Statement	$a	$x	y
1	`$a = trim(fgets(STDIN))`	**4**	?	?
2	`$x = trim(fgets(STDIN))`	4	**6**	?
3	`$y = trim(fgets(STDIN))`	4	6	**2**
4	`case $a == 10`	False		
5	`case $a == 3`	False		
6	`case $a == 5`	False		
7	`$x -= 3`	4	**3**	2
8	`$y++`	4	3	**3**
9	`echo $x, $y`	3 3 is displayed		

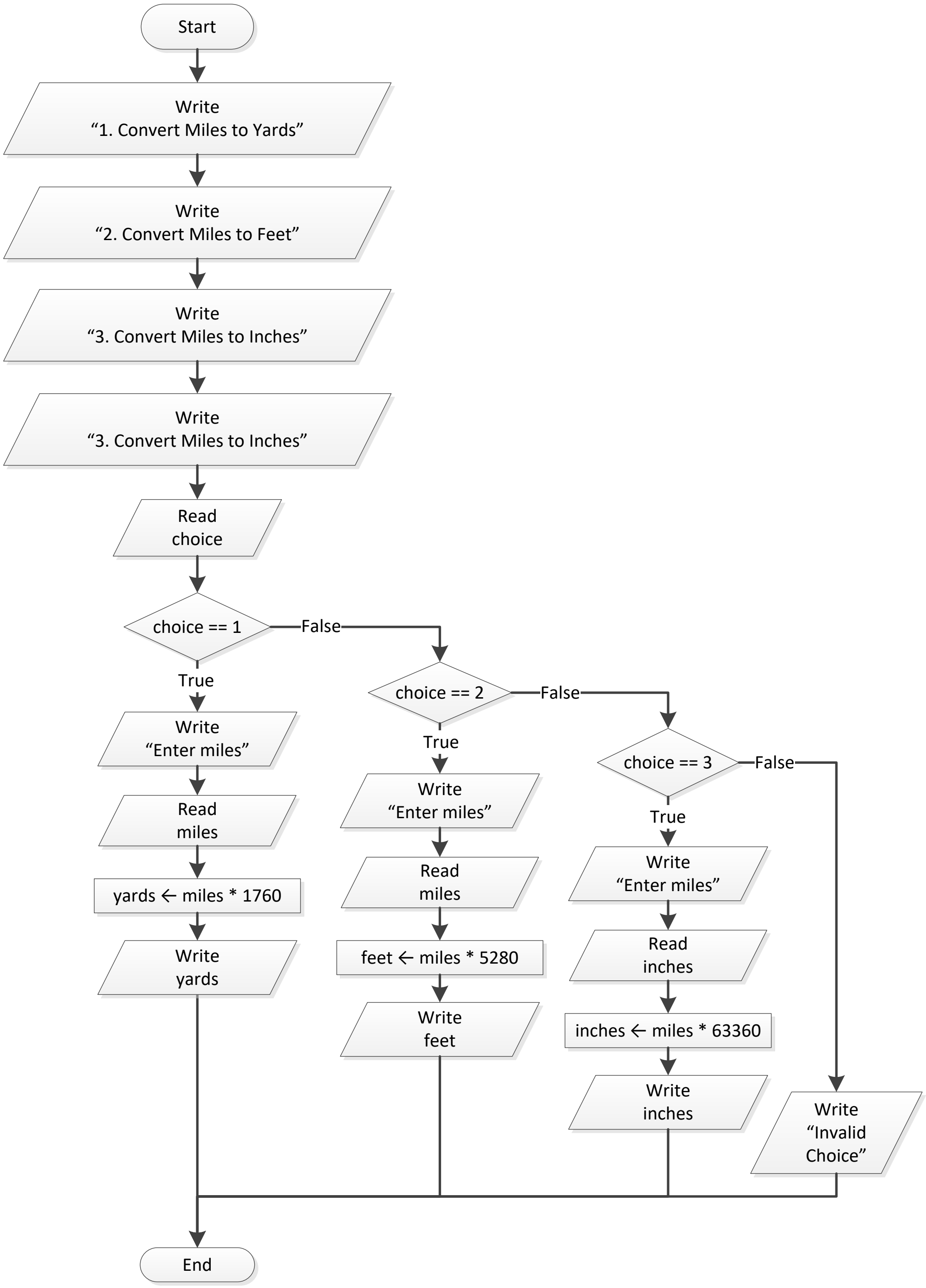


3. *Solution*

```
<?php
  echo "Enter the name of a month: ";
  $name = trim(fgets(STDIN));

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

```
?>
```

4. Solution



```
<?php
  echo "1. Convert Miles to Yards\n";
  echo "2. Convert Miles to Feet\n";
  echo "3. Convert Miles to Inches\n";

  echo "Enter a choice: ";
  $choice = trim(fgets(STDIN));

  switch ($choice) {
    case 1:
      echo "Enter miles: ";
      $miles = trim(fgets(STDIN));
      $yards = $miles * 1760;
      echo $miles, " miles = " , $yards, " yards";
      break;
    case 2:
      echo "Enter miles: ";
      $miles = trim(fgets(STDIN));
      $feet = $miles * 5280;
      echo $miles, " miles = " , $feet, " feet";
      break;
    case 3:
      echo "Enter miles: ";
      $miles = trim(fgets(STDIN));
      $inches = $miles * 63360;
      echo $miles, " miles = " , $inches, " inches";
      break;
    default:
      echo "Invalid choice!";
  }
?>
```

5. *Solution*

```
<?php
  echo "Enter a Roman numeral: ";
  $roman = trim(fgets(STDIN));

  switch (strtoupper($roman)) {
    case "I":
      echo "1";
      break;
    case "II":
      echo "2";
      break;
    case "III":
      echo "3";
      break;
    case "IV":
      echo "4";
      break;
    case "V":
      echo "5";
      break;
    case "VI":
      echo "6";
      break;
    case "VII":
      echo "7";
      break;
    case "VIII":
      echo "8";
      break;
    case "IX":
      echo "9";
      break;
    case "X":
      echo "10";
      break;
    default:
      echo "Error";
  }
?>
```

6. *Solution*

```
<?php
  echo "Enter the total number of CDs purchased in a month: ";
  $total = trim(fgets(STDIN));

  switch ($total) {
    case 1:
      echo "You are awarded 3 points";
      break;
    case 2:
      echo "You are awarded 10 points";
      break;
    case 3:
      echo "You are awarded 20 points";
      break;
    default:
      echo "You are awarded 45 points";
  }
?>
```

7. Solution

```php
<?php
  echo "Enter your name: ";
  $name = trim(fgets(STDIN));

  $i = rand(1, 3);

  switch ($i) {
    case 1:
      echo "Good morning " , $name;
      break;
    case 2:
      echo "Good evening " , $name;
      break;
    case 3:
      echo "Good night " , $name;
      break;
  }
?>
```

8. Solution

```php
<?php
  $num_string = trim(fgets(STDIN));

  switch (strtoupper($num_string)) {
    case "ZERO":
      echo 0;
      break;
    case "ONE":
      echo 1;
      break;
    case "TWO":
      echo 2;
      break;
    case "THREE":
      echo 3;
      break;
    case "FOUR":
      echo 4;
      break;
    case "FIVE":
      echo 5;
      break;
    case "SIX":
      echo 6;
      break;
    case "SEVEN":
      echo 7;
      break;
    case "EIGHT":
      echo 8;
      break;
    case "NINE":
      echo 9;
      break;
    default:
      echo "I don't know this number!";
  }
?>
```

9. Solution

```
<?php
  echo "Enter Beaufort number: ";
  $b = trim(fgets(STDIN));

  switch ($b) {
    case 0:
      echo "Calm";
      break;
    case 1:
      echo "Light Air";
      break;
    case 2:
      echo "Light breeze";
      break;
    case 3:
      echo "Gentle breeze";
      break;
    case 4:
      echo "Moderate breeze";
      break;
    case 5:
      echo "Fresh breeze";
      break;
    case 6:
      echo "Strong breeze";
      break;
    case 7:
      echo "Moderate gale";
      break;
    case 8:
      echo "Gale";
      break;
    case 9:
      echo "Strong gale";
      break;
    case 10:
      echo "Storm";
      break;
    case 11:
      echo "Violent storm";
      break;
    case 12:
      echo "Hurricane force";
      break;
    default:
      echo "Invalid Beaufort number!";
  }
?>
```

Chapter 20

20.3 Answers of Review Questions: True/False

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

20.4 Answers of Review Exercises

1. Solution

For input values of 20, 1

Step	Statement	$x	$y
1	`$x = trim(fgets(STDIN))`	**20**	?
2	`$y = trim(fgets(STDIN))`	20	**1**
3	`if ($x < 30)`	True	
4	`case $y == 1`	True	
5	`$x = $x % 3`	**2**	1
6	`$y = 5`	2	**5**
7	`echo $x, $y`	2 5 is displayed	

For input values of 20, 3

Step	Statement	$x	$y
1	`$x = trim(fgets(STDIN))`	**20**	?
2	`$y = trim(fgets(STDIN))`	20	**3**
3	`if ($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	`echo $x, $y`	25 6 is displayed	

For input values of 12, 8

Step	Statement	$x	$y
1	`$x = trim(fgets(STDIN))`	**12**	?
2	`$y = trim(fgets(STDIN))`	12	**8**
3	`if ($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	`echo $x, $y`	10 9 is displayed	

For input values of 50, 0

Step	Statement	$x	$y
1	`$x = trim(fgets(STDIN))`	**50**	?
2	`$y = trim(fgets(STDIN))`	50	**0**
3	`$y++`	50	**1**
4	`echo $x, $y`	50 1 is displayed	

2. *Solution*

For input values of 60, 25

Step	Statement	$x	$y
1	`$x = trim(fgets(STDIN))`	**60**	?
2	`$y = trim(fgets(STDIN))`	60	25
3	`if (($x + $y) / 2 <= 20)`	False	
4	`if ($y < 15)`	False	
5	`elseif ($y < 23)`	False	
6	`$x = 2 * $x + 5`	**125**	25
7	`$y += 1`	125	**26**
8	`echo $x, $y`	125 26 is displayed	

For input values of 50, 8

Step	Statement	$x	$y
1	`$x = trim(fgets(STDIN))`	**50**	?
2	`$y = trim(fgets(STDIN))`	50	8
3	`if (($x + $y) / 2 <= 20)`	False	
4	`if ($y < 15)`	True	
5	`$x = $x % 4`	**2**	8
6	`$y = 2`	2	**2**
7	`echo $x, $y`	2 2 is displayed	

For input values of 20, 15

Step	Statement	$x	$y
1	`$x = trim(fgets(STDIN))`	**20**	?
2	`$y = trim(fgets(STDIN))`	20	15
3	`if (($x + $y) / 2 <= 20)`	True	

4	`if ($y < 10)`	False	
5	`elseif ($y < 20)`	True	
6	`$x = $x * 5`	**100**	15
7	`$y += 2`	100	**17**
8	`echo $x, $y`	100 17 is displayed	

For input values of 10, 30

Step	Statement	$x	$y
1	`$x = trim(fgets(STDIN))`	**10**	?
2	`$y = trim(fgets(STDIN))`	10	30
3	`if (($x + $y) / 2 <= 20)`	True	
4	`if ($y < 10)`	False	
5	`elseif ($y < 20)`	False	
6	`$x = $x - 2`	**8**	30
7	`$y += 3`	8	**33**
8	`echo $x, $y`	8 33 is displayed	

3. Solution

```php
<?php
  echo "Enter the three sides of a triangle: ";
  $a = trim(fgets(STDIN));
  $b = trim(fgets(STDIN));
  $c = trim(fgets(STDIN));

  if ($a >= $b + $c || $b >= $a + $c || $c >= $a + $b) {
    echo "Given numbers cannot be lengths of the three sides of a triangle";
  }
  else {
    if ($a == $b && $b == $c) {
      echo "Equilateral";
    }
    elseif (pow($a, 2) == pow($b, 2) + pow($c, 2) ||
            pow($b, 2) == pow($a, 2) + pow($c, 2) ||
            pow($c, 2) == pow($a, 2) + pow($b, 2)) {
      echo "Right (or right-angled)";
    }
    else {
      echo "Not special";
    }
  }
?>
```

4. Solution

```
<?php
  echo "Enter your four-digit PIN : ";
  $pin= trim(fgets(STDIN));
  if ($pin != "1234") {
    echo "Wrong PIN. Enter your four-digit PIN : ";
    $pin = trim(fgets(STDIN));
    if ($pin != "1234") {
      echo "Wrong PIN. Enter your four-digit PIN : ";
      $pin = trim(fgets(STDIN));
    }
  }

  if ($pin != "1234") {
    echo "PIN locked!";
  }
  else {
    echo "Enter the amount of money (an integer value) that you want to withdraw: ";
    $amount = trim(fgets(STDIN));
    $usd10 = intval($amount / 10);
    $r = $amount % 10;
    $usd5 = intval($r / 5);
    $usd1 = $r % 5;
    echo $usd10, " notes of $10 ", $usd5 , " notes of $5, and ", $usd1, " notes of $1";
  }
?>
```

5. Solution

First Approach

```
<?php
  echo "Enter temperature (in Fahrenheit): ";
  $t = trim(fgets(STDIN));
  echo "Enter wind speed (in miles/hour): ";
  $w = trim(fgets(STDIN));

  if ($t > 75) {
    if ($w > 12) {
      echo "The day is hot and windy";
    }
    else {
      echo "The day is hot and not windy";
    }
  }
  else {
    if ($w > 12) {
      echo "The day is cold and windy";
    }
    else {
      echo "The day is cold and not windy";
    }
  }
?>
```

Second Approach

```
<?php
  echo "Enter temperature (in Fahrenheit): ";
  $t = trim(fgets(STDIN));
  echo "Enter wind speed (in miles/hour): ";
  $w = trim(fgets(STDIN));

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

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

  echo "The day is ", $message1, " and ", $message2;
?>
```

Chapter 21

21.13 Answers of 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. false
13. false

21.14 Answers of Review Questions: Multiple Choice

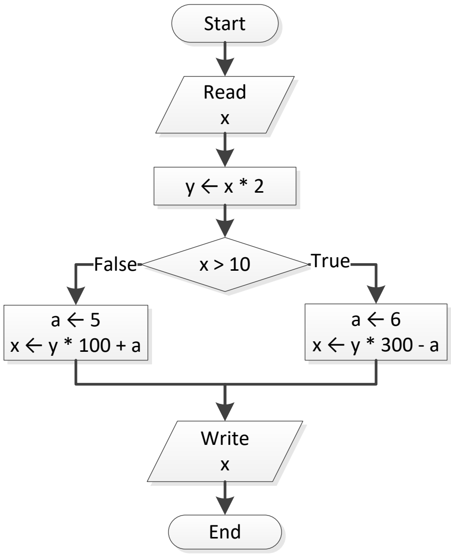
1. c
2. b
3. a
4. a
5. b
6. c
7. c
8. d

21.15 Answers of Review Exercises

1. Solution

```
<?php
  $y = trim(fgets(STDIN));
  $x = trim(fgets(STDIN));

  if ($y > 0) {
    $a = $x * 4 * $y + 1;
  }
  else {
    $a = $x * 2 * $y + 6;
  }
  echo $y;
  echo $a;
?>
```

2. Solution



3. Solution

```php
<?php
  $a = trim(fgets(STDIN));

  if ($a >= 10) {
    echo "Error!";
  }
  else {
    if ($a < 1) {
      $y = 5 + $a;
    }
    elseif ($a < 5) {
      $y = 23 / $a;
    }
    else {
      $y = 5 * $a;
    }
    echo $y;
  }
?>
```

4. Solution

```php
<?php
  $day = trim(fgets(STDIN));
  $month = trim(fgets(STDIN));
  $name = trim(fgets(STDIN));
  if ($day == 16 && $month == 2 && $name == "Loukia") {
    echo "Happy Birthday!!!";
  }
  else {
    echo "No match!";
  }
?>
```

5. Solution

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

```
<?php
  $a = trim(fgets(STDIN));
  $b = trim(fgets(STDIN));
  $c = trim(fgets(STDIN));

  if ($a > 10) {
    if ($c < 2000) {
      $d = ($a + $b + $c) / 12;
      echo "The result is: ", $d;
    }
    else {
      echo "Error!";
    }
  }
  else {
    echo "Error!";
  }
?>
```

6. Solution

```
<?php
  $a = trim(fgets(STDIN));
  $b = trim(fgets(STDIN));
  $c = trim(fgets(STDIN));

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

  if ($a <= 10) {
    echo "Error!";
  }
?>
```

7. Solution

```php
<?php
  $a = trim(fgets(STDIN));
  $b = trim(fgets(STDIN));

  $y = 3;
  if ($a > 0) {
    $y = $y * $a;
    echo "Hello Zeus";
  }

  echo $y, $b;
?>
```

8. Solution

```php
<?php
  $a = trim(fgets(STDIN));
  $b = trim(fgets(STDIN));

  $y = 0;
  if ($a > 0) {
    $y = $y + 7;
  }
  else {
    echo "Hello Zeus";
    echo abs($a);
  }
  echo $y;
?>
```

9. Solution

```php
<?php
  echo "What is your tablet's OS? ";
  $os = trim(fgets(STDIN));

  if ($os == "iOS" ) {
    echo "Apple";
  }
  elseif ($os == "Android") {
    echo "Google";
  }
  elseif ($os == "Windows") {
    echo "Microsoft";
  }
```

```
?>
```

10. Solution

```
<?php
  $a = trim(fgets(STDIN));
  $x = trim(fgets(STDIN));
  $y = trim(fgets(STDIN));

  if ($a == 3) {
    $x = $x / 4;
    $y = pow($y, 5);
  }
  elseif ($a == 7) {
    $x = $x * 3;
    $y++;
  }
  elseif ($a == 22) {
    $x = $x % 4;
    $y += 9;
  }
  else {
    $x -= 9;
    $y++;
  }

  echo $x, $y;
?>
```

11. Solution

```
<?php
  $a = trim(fgets(STDIN));
  $x = trim(fgets(STDIN));
  $y = trim(fgets(STDIN));

  if($a == 3) {
    $x = $x / 4;
    $y = pow($y, 5);
  }
  else {
    if ($a == 7) {
      $x = $x * 3;
      $y++;
    }
    else {
      if ($a == 22) {
        $x = $x % 4;
        $y += 9;
      }
      else {
        $x -= 9;
        $y++;
      }
    }
  }

  echo $x, $y;
?>
```

12. Solution

```
<?php
  echo "Enter a color in hexadecimal: ";
  $color = trim(fgets(STDIN));

  switch ($color) {
    case "FF0000":
      echo "red";
      break;
    case "00FF00":
      echo "green";
      break;
    case "0000FF":
      echo "blue";
      break;
    case "FFFFFF":
      echo "white";
      break;
    case "000000":
      echo "black";
      break;
    case "7F7F7F":
      echo "gray";
      break;
    default:
      echo "Unknown color!";
  }
?>
```

13. Solution

```
<?php
  echo "Enter a color in hexadecimal: ";
  $color = trim(fgets(STDIN));

  if ($color == "FF0000") {
    echo "red";
  }
  else {
    if ($color == "00FF00") {
      echo "green";
    }
    else {
      if ($color == "0000FF") {
        echo "blue";
      }
      else {
        if ($color == "FFFFFF") {
          echo "white";
        }
        else {
          if ($color == "000000") {
            echo "black";
          }
          else {
            if ($color == "7F7F7F") {
              echo "gray";
            }
            else {
              echo "Unknown color!";
            }
          }
        }
      }
    }
  }
?>
```

14. Solution

```
<?php
  $a = trim(fgets(STDIN));

  if ($a > 1000)
    echo "Big Positive";
  else {
    if ($a > 0)
      echo "Positive";
    else {
      if ($a < -1000)
```

```
        echo "Big Negative";
      else {
        if ($a < 0)
          echo "Negative";
        else
          echo "Zero";
      }
    }
  }
?>
```

```
<?php

  $a = trim(fgets(STDIN));

  if ($a > 1000)
    echo "Big Positive";
  elseif ($a > 0)
    echo "Positive";
  elseif ($a < -1000)
    echo "Big Negative";
  elseif ($a < 0)
    echo "Negative";
  else
    echo "Zero";
?>
```

15. Solution

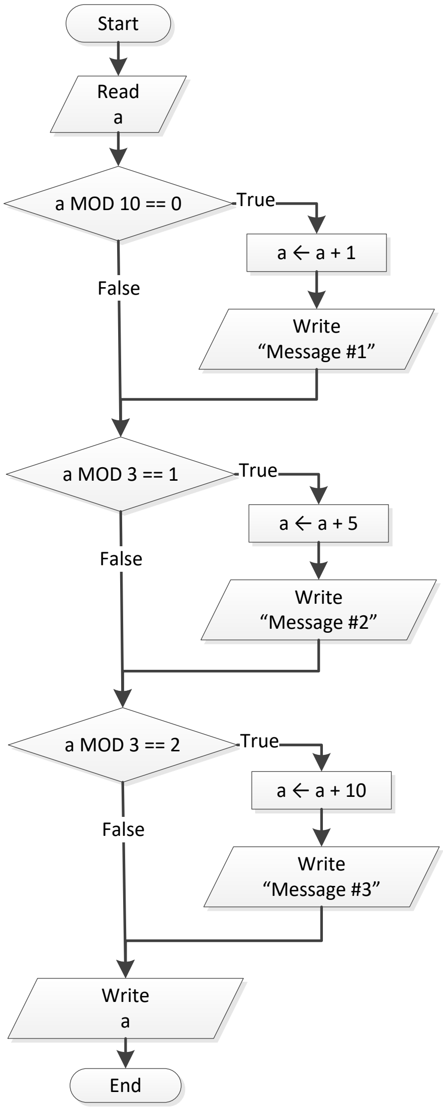
```
<?php
  $a = trim(fgets(STDIN));

  if ($a < 1) {
    $y = 5 + $a;
    echo $y;
  }
  elseif ($a < 5) {
    $y = 23 / $a;
    echo $y;
  }
  elseif ($a < 10) {
    $y = 5 * $a;
    echo $y;
  }
  else {
    echo "Error!";
  }
?>
```

```
<?php
  $a = trim(fgets(STDIN));

  if ($a < 1) {
    $y = 5 + $a;
    echo $y;
  }
  else {
    if ($a < 5) {
      $y = 23 / $a;
      echo $y;
    }
    else {
      if ($a < 10) {
        $y = 5 * $a;
        echo $y;
      }
      else {
        echo "Error!";
      }
    }
  }
?>
```

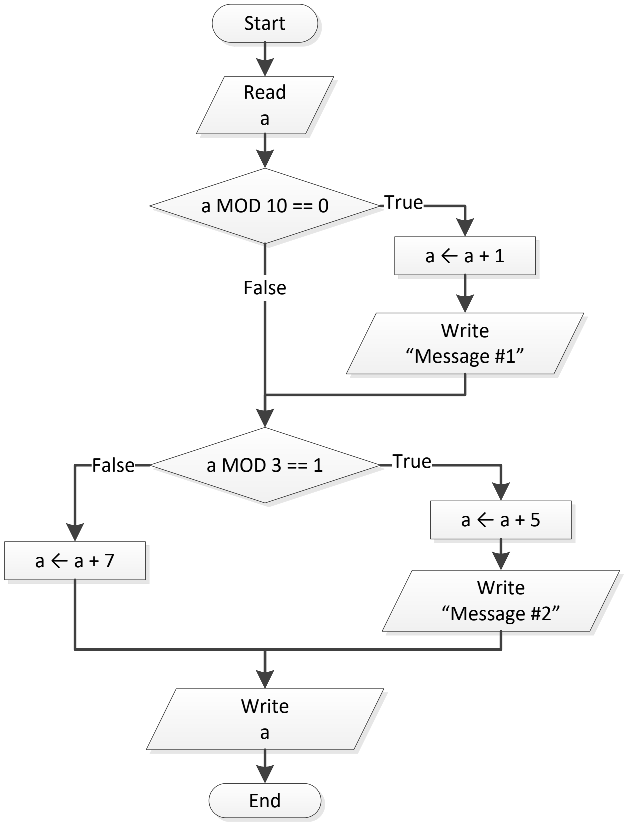
Chapter 22

22.4 Answers of Review Exercises

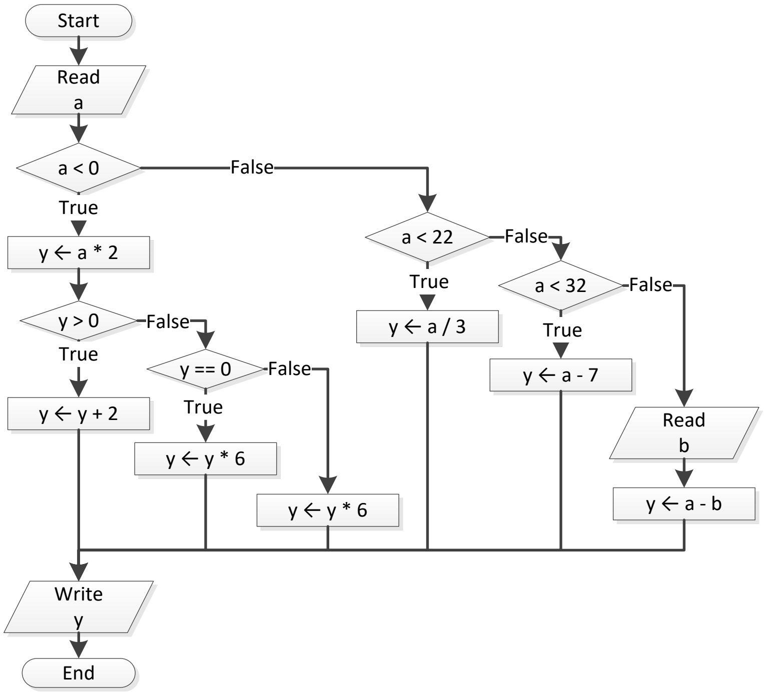
1. Solution



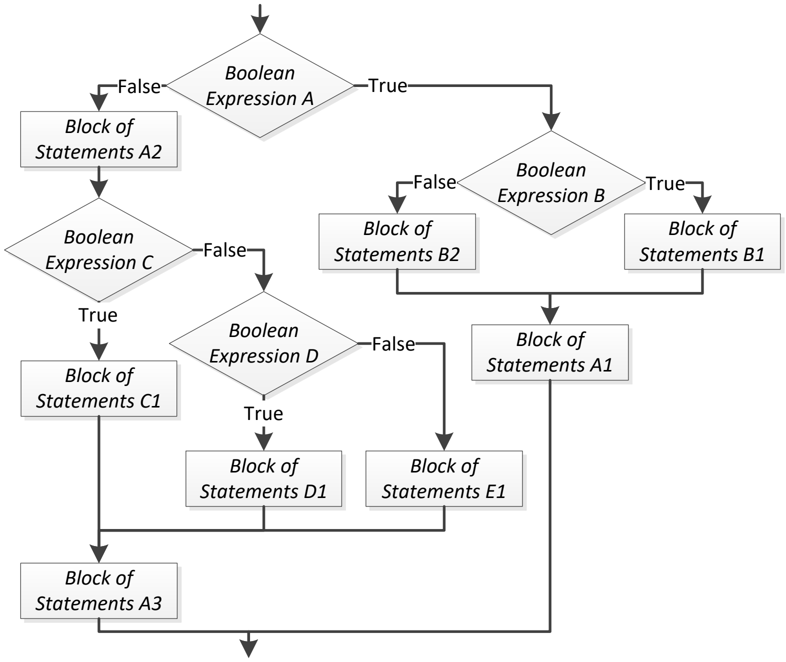
2. Solution



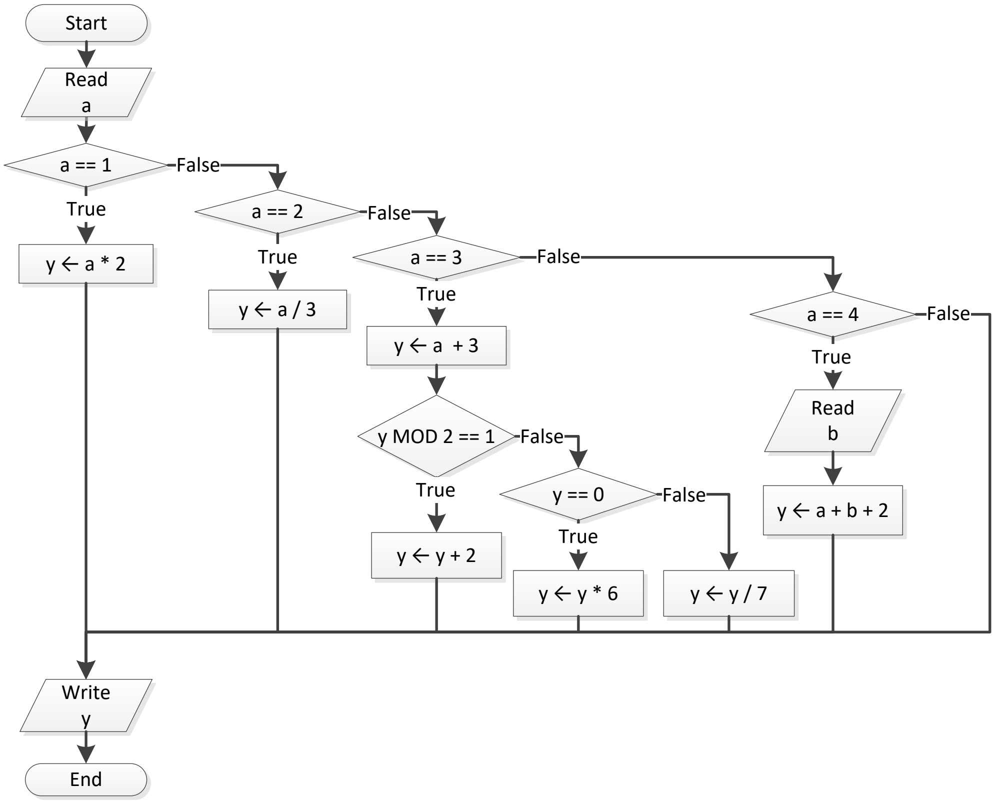
3. Solution



4. Solution



5. Solution



6. Solution

```php
<?php
  $x = trim(fgets(STDIN));
  $y = trim(fgets(STDIN));

  if ($x != 100 || $y <= 10) {
    $z = trim(fgets(STDIN));
    if ($z <= $x + $y) {
      $x -= 3;
      $y = $x + 4;
    }
  }
  echo $x, $y;
?>
```

7. Solution

```php
<?php
  $x = trim(fgets(STDIN));

  if ($x == 1) {
    echo "Good Morning";
    echo "How do you do?";
    echo "Is everything okay?";
  }
  elseif ($x == 2) {
    echo "Good Evening";
    echo "How do you do?";
    echo "Is everything okay?";
  }
  elseif ($x == 3) {
    echo "Good Afternoon";
    echo "Is everything okay?";
  }
  else {
    echo "Good Night";
  }
?>
```

8. Solution

```php
<?php
  $x = trim(fgets(STDIN));

  if (is_numeric($x) == true) {
    if ($x % 10 == 0) {
      echo "Last digit equal to 0";
    }
    elseif ($x % 10 == 1) {
      echo "Last digit equal to 1";
    }
    else {
      echo "None";
    }
  }
  else {
    if ($x == "Exit") {
      echo "Bye";
    }
    else {
      echo "Invalid Number";
    }
  }
?>
```

9. Solution

```
<?php
  $a = trim(fgets(STDIN));
  $b = trim(fgets(STDIN));

  $y = $a * $b;

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

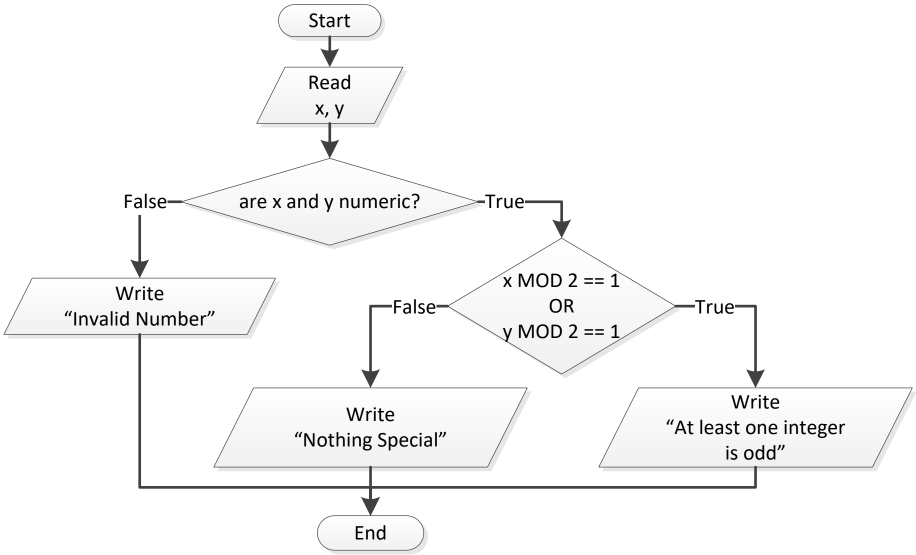
10. Solution

```
<?php
  $a = trim(fgets(STDIN));
  $b = trim(fgets(STDIN));
  $c = trim(fgets(STDIN));

  $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;
    }
  }
  echo $a, $b, $c;
?>
```

Chapter 23

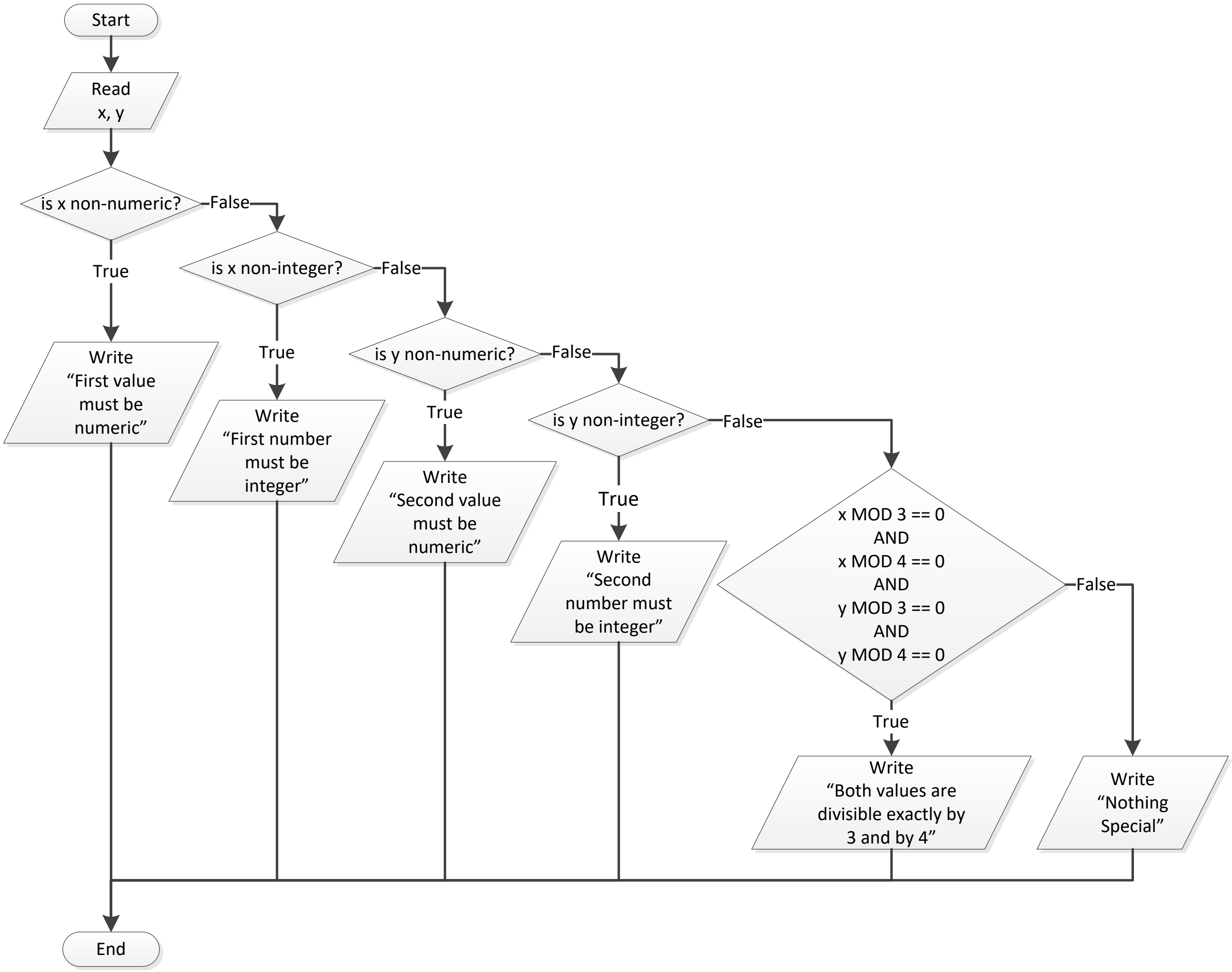
23.6 Answers of Review Exercises

1. *Solution*



```
<?php
  $x = trim(fgets(STDIN));
  $y = trim(fgets(STDIN));

  if (is_numeric($x) == true && is_numeric($y) == true) {
    if ($x % 2 == 1 || $y % 2 == 1) {
      echo "At least one integer is odd";
    }
    else {
      echo "Nothing Special";
    }
  }
  else {
    echo "Invalid Number";
  }
?>
```

2. *Solution*


Start
Read x, y
is x non-numeric?
False
True
Write "First value must be numeric"
is x non-integer?
False
True
Write "First number must be integer"
is y non-numeric?
False
True
Write "Second value must be numeric"
is y non-integer?
False
True
Write "Second number must be integer"
x MOD 3 == 0 AND x MOD 4 == 0 AND y MOD 3 == 0 AND y MOD 4 == 0
False
True
Write "Both values are divisible exactly by 3 and by 4"
Write "Nothing Special"
End

```
<?php
  $x = trim(fgets(STDIN));
  $y = trim(fgets(STDIN));

  if (is_numeric($x) != true) {
    echo "First value must be numeric";
  }
  elseif ($x != intval($x)) {
    echo "First number must be integer";
  }
  elseif (is_numeric($y) != true) {
    echo "Second value must be numeric";
  }
  elseif ($y != intval($y)) {
    echo "Second number must be integer";
  }
  elseif ($x % 3 == 0 && $x % 4 == 0 && $y % 3 == 0 && $y % 4 == 0 ) {
    echo "Both values are divisible exactly by 3 and by 4";
  }
  else {
    echo "Nothing Special";
  }
?>
```

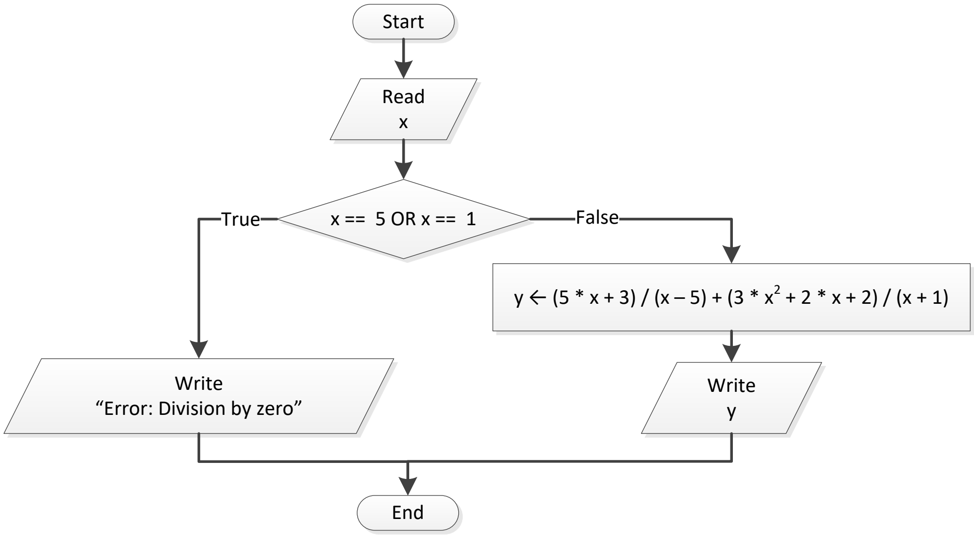
3. Solution

```
<?php
  echo "1. Convert Kelvin to Fahrenheit\n";
  echo "2. Convert Fahrenheit to Kelvin\n";
  echo "3. Convert Fahrenheit to Celsius\n";
  echo "4. Convert Celsius to Fahrenheit\n";

  echo "Enter a choice: ";
  $choice = trim(fgets(STDIN));
  echo "Enter a temperature: ";
  $t = trim(fgets(STDIN));

  if ($choice < 1 || $choice > 4) {
    echo "Wrong choice";
  }
  elseif (is_numeric($t) != true) {
    echo "Wrong temperature";
  }
  else {
    switch ($choice) {
      case 1:
        echo 1.8 * $t - 459.67;
        break;
      case 2:
        echo ($t + 459.57) / 1.8;
        break;
      case 3:
        echo 5 / 9 * ($t - 32);
        break;
      case 4:
        echo 9 / 5 * $t + 32;
        break;
    }
  }
?>
```

4. Solution

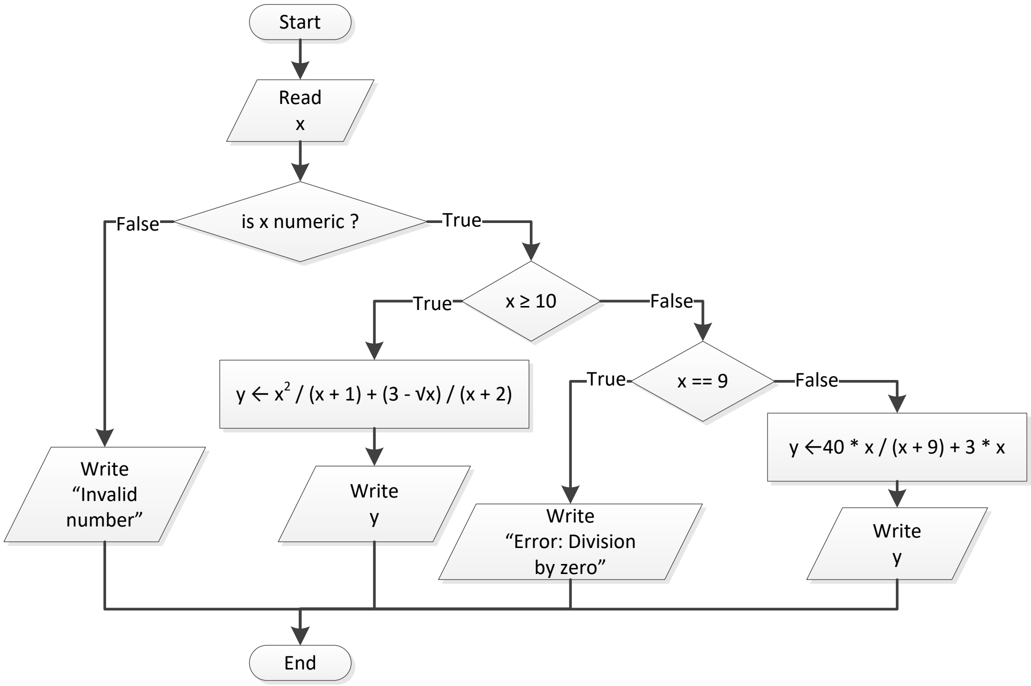
```
<?php
  echo "Enter 1st integer: ";
  $a = trim(fgets(STDIN));
  echo "Enter type of operation: ";
  $op = trim(fgets(STDIN));
  echo "Enter 2nd integer: ";
  $b = trim(fgets(STDIN));

  switch ($op) {
    case "+":
      echo $a + $b;
      break;
    case "-":
      echo $a - $b;
      break;
    case "*":
      echo $a * $b;
      break;
    case "/":
      if ($b == 0) {
        echo "Error: Division by zero";
      }
      else {
        echo $a / $b;
      }
      break;
    case "DIV":
      if ($b == 0) {
        echo "Error: Division by zero";
      }
      else {
        echo intval($a / $b);
      }
      break;
    case "MOD":
      if ($b == 0) {
        echo "Error: Division by zero";
      }
      else {
        echo $a % $b;
      }
      break;
    case "POWER":
      echo pow($a, $b);
      break;
  }
?>
```

5. Solution


Start
Read
x
x == 5 OR x == 1
True
False
y ← (5 * x + 3) / (x – 5) + (3 * x^2 + 2 * x + 2) / (x + 1)
Write
"Error: Division by zero"
Write
y
End

```
<?php
  $x = trim(fgets(STDIN));

  if ($x == 5 || $x == 1) {
    echo "Error: Division by zero";
  }
  else {
    $y = (5 * x + 3) / (x - 5) + (3 * pow($x, 2) + 2 * x + 2) / (x + 1);
    echo $y;
  }
?>
```

6. Solution



```
<?php
  $x = trim(fgets(STDIN));

  if (is_numeric(x) == true) {
    if ($x >= 10) {
      $y = pow($x, 2) / ($x + 1) + (3 - sqrt($x)) / ($x + 2);
      echo $y;
    }
    else {
      if ($x == 9) {
        echo "Error: Division by zero";
      }
      else {
        $y = 40 * $x / ($x + 9) + 3 * $x;
        echo $y;
      }
    }
  }
  else {
    echo "Invalid number";
  }
?>
```

7. Solution

```
<?php
```

```
  $x = trim(fgets(STDIN));

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

8. Solution

```
<?php
  echo "Enter age for person No1:";
  $age1 = trim(fgets(STDIN));
  echo "Enter age for person No2:";
  $age2 = trim(fgets(STDIN));
  echo "Enter age for person No3:";
  $age3 = trim(fgets(STDIN));

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

  $middle = $age1 + $age2 + $age3 - $min - $max;
  echo $middle;
?>
```

9. Solution

```
<?php
  echo "Enter the age of the first person: ";
  $a1 = trim(fgets(STDIN));
  echo "Enter the name of the first person: ";
  $n1 = trim(fgets(STDIN));
  echo "Enter the age of the second person: ";
  $a2 = trim(fgets(STDIN));
  echo "Enter the name of the second person: ";
  $n2 = trim(fgets(STDIN));
  echo "Enter the age of the third person: ";
  $a3 = trim(fgets(STDIN));
  echo "Enter the name of the third person: ";
  $n3 = trim(fgets(STDIN));

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

  $max = $a1;
  $max_name = $n1;
  if ($a2 > $max) {
    $max = $a2;
    $max_name = $n2;
  }
  if ($a3 > $max) {
    $max = $a3;
    $max_name = $n3;
  }
  $middle = $a1 + $a2 + $a3 - $min - $max;

  if (abs($max - $middle) < abs($min - $middle)) {
    echo $max_name;
  }
  else {
    echo $min_name;
  }

?>
```

10. Solution

```
<?php
  echo "Enter a three-digit integer: ";
  $x = trim(fgets(STDIN));

  if (is_numeric($x) != true) {
    echo "Entered value contains non-numeric characters";
  }
  elseif ($x < 100 || $x > 999) {
    echo "Entered integer is not a three-digit integer";
  }
  else {
    $digit1 = intval($x / 100);
    $r = $x % 100;

    $digit2 = intval($r / 10);
    $digit3 = $r % 10;

    $sum = pow($digit1, 3) + pow($digit2, 3) + pow($digit3, 3);

    if ($sum == $x) {
      echo "You entered an Armstrong number!";
    }
    else {
      echo "You entered a non-Armstrong number!";
    }

  }
?>
```

11. Solution

```
<?php
  echo "Enter day 1 - 31: ";
  $d = trim(fgets(STDIN));
  echo "Enter month 1 - 12: ";
  $m = trim(fgets(STDIN));
  echo "Enter year: ";
  $y = trim(fgets(STDIN));

  if ($m == 2) {
    if ($y % 4 == 0 && $y % 100 != 0 || $y % 400 == 0) {
      echo 29 - $d;
    }
    else {
      echo 28 - $d;
    }
  }
  elseif ($m == 4 || $m == 6 || $m == 9 || $m == 11) {
    echo 30 - $d;
  }
  else {
    echo 31 - $d;
  }
?>
```

12. Solution

```
<?php
  $word = trim(fgets(STDIN));

  if ($word[0] == strtolower($word[0]) &&
      $word[1] == strtoupper($word[1]) &&
      $word[2] == strtolower($word[2]) &&
      $word[3] == strtoupper($word[3]) &&
      $word[4] == strtolower($word[4]) &&
      $word[5] == strtoupper($word[5]) ||
      $word[0] == strtoupper($word[0]) &&
      $word[1] == strtolower($word[1]) &&
      $word[2] == strtoupper($word[2]) &&
      $word[3] == strtolower($word[3]) &&
      $word[4] == strtoupper($word[4]) &&
      $word[5] == strtolower($word[5])) {
    echo "Word is okay!";
  }
  else {
    echo "Word is not okay";
  }

?>
```

13. Solution

```php
<?php
  echo "Enter quantity: ";
  $q = trim(fgets(STDIN));

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

  $payment = $q * 10 - $q * 10 * $discount / 100;

  echo "You got a discount of ", $discount, "%\n";
  echo "You must pay $", $payment;
?>
```

14. Solution

```php
<?php
  define("VAT", 0.19);

  echo "Enter total amount: ";
  $amount = trim(fgets(STDIN));

  if (is_numeric($amount) != true) {
    echo "Entered value contains non-numeric characters";
  }
  elseif ($amount < 0) {
    echo "Entered non-negative value";
  }
  else {
    if ($amount < 50) {
      $discount = 0;
    }
    elseif ($amount < 100) {
      $discount = 1;
    }
    elseif ($amount < 250) {
      $discount = 2;
    }
    else {
      $discount = 3;
    }

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

    echo "You got a discount of ", $discount, "%\n";
    echo "You must pay $", $payment;
  }
?>
```

15. Solution

```
<?php
  echo "Enter age: ";
  $a = trim(fgets(STDIN));
  if ($a < 18) {
    echo "Invalid age";
  }
  else {
    echo "Enter weight in pounds: ";
    $w = trim(fgets(STDIN));
    echo "Enter height in inches: ";
    $h = trim(fgets(STDIN));

    $bmi = $w * 703 / ($h * $h);

    if ($bmi < 15) {
      echo "Very severely underweight";
    }
    elseif ($bmi < 16) {
      echo "Severely underweight";
    }
    elseif ($bmi < 18.5) {
      echo "Underweight";
    }
    elseif ($bmi < 25) {
      echo "Normal";
    }
    elseif ($bmi < 30) {
      echo "Overweight";
    }
    elseif ($bmi < 35) {
      echo "Severely overweight";
    }
    else {
      echo "Very severely overweight";
    }
  }
?>
```

16. Solution

```
<?php
  define("TAX_RATE", 0.10);

  echo "Enter water consumption (in cubic feet): ";
  $water = trim(fgets(STDIN));

  if (is_numeric($water) != true) {
    echo "Entered value contains non-numeric characters";
  }
  elseif ($water < 0) {
    echo "Entered value is negative";
  }
  else {
    if ($water <= 10) {
      $total = $water * 3;
    }
    elseif ($water <= 20) {
      $total = 10 * 3 + ($water - 10) * 5;
    }
    elseif ($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;
    echo "Total amount to pay (taxes included): ", $total;
  }
?>
```

17. Solution

```
<?php
  echo "Enter taxable income: ";
  $income = trim(fgets(STDIN));
  echo "Enter number of children: ";
  $children = trim(fgets(STDIN));

  if ($income <= 8000) {
    $tax = $income * 0.10;
  }
  elseif ($income <= 30000) {
    $tax = 8000 * 0.10 + ($income - 8000) * 0.15;
  }
  elseif ($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;
  }
  echo "Tax: ", $tax;
?>
```

18. Solution

```
<?php

  echo "Enter wind speed (in miles/hour): ";
  $wind = trim(fgets(STDIN));

  if (is_numeric($wind) != true) {
    echo "Entered value contains non-numeric characters";
  }
  elseif ($wind < 0) {
    echo "Entered value is negative";
  }
  else {
    if ($wind < 1) {
      echo "Beaufort: 0\nCalm";
    }
    elseif ($wind < 4) {
      echo "Beaufort: 1\nLight air";
    }
    elseif ($wind < 8) {
      echo "Beaufort: 2\nLight breeze";
    }
    elseif ($wind < 13) {
      echo "Beaufort: 3\nGentle breeze";
    }
    elseif ($wind < 18) {
      echo "Beaufort: 4\nModerate breeze";
    }
    elseif ($wind < 25) {
      echo "Beaufort: 5\nFresh breeze";
    }
    elseif ($wind < 31) {
      echo "Beaufort: 6\nStrong breeze";
    }
    elseif ($wind < 39) {
      echo "Beaufort: 7\nModerate gale";
    }
    elseif ($wind < 47) {
      echo "Beaufort: 8\nGale";
    }
    elseif ($wind < 55) {
      echo "Beaufort: 9\nStrong gale";
    }
    elseif ($wind < 64) {
      echo "Beaufort: 10\nStorm";
    }
    elseif ($wind < 74) {
      echo "Beaufort: 11\nViolent storm";
    }
    else {
      echo "Beaufort: 12\nHurricane force";
    }

    if ($wind < 13) {
      echo "It's Fishing Day!!!";
    }
  }
?>
```

Chapter 24

24.3 Answers of Review Questions: True/False

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

Chapter 25

25.2 Answers of Review Questions: True/False

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

25.3 Answers of Review Questions: Multiple Choice

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

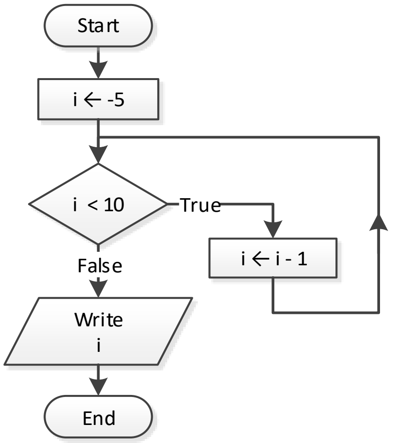
25.4 Answers of Review Exercises

1. Solution

Step	Statement	$i	$x
1	`$i = 3`	**3**	?
2	`$x = 0`	3	**0**
3	`while ($i >= 0)`	True	
4	`$i--`	**2**	0
5	`$x += $i`	2	**2**
6	`while ($i >= 0)`	True	
7	`$i--`	**1**	2
8	`$x += $i`	1	**3**
9	`while ($i >= 0)`	True	
10	`$i--`	**0**	3
11	`$x += $i`	0	**3**
12	`while ($i >= 0)`	True	
13	`$i--`	**-1**	3
14	`$x += $i`	-1	**2**
15	`while ($i >= 0)`	False	
16	`echo $x`	2 is displayed	

It performs 3 iterations

2. Solution



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

It performs an infinite number of iterations

3. Solution

Step	Statement	$a	$b	$c	$d
1	`$a = 2`	**2**	?	?	?
2	`while ($a <= 10)`	True			
3	`$b = $a + 1`	2	**3**	?	?
4	`$c = $b * 2`	2	3	**6**	?
5	`$d = $c - $b + 1`	2	3	6	**4**
6	`$d == 4`	True			
7	`echo $b, $c`	3 6 is displayed			
8	`$a += 4`	**6**	3	6	4
9	`while ($a <= 10)`	True			
10	`$b = $a + 1`	6	**7**	6	4
11	`$c = $b * 2`	6	7	**14**	4

12	`$d = $c - $b + 1`	6	7	14	**8**
13	`$d == 4`	False			
14	`$d == 5`	False			
15	`$d == 8`	True			
16	`echo $a, $b`	6 7 is displayed			
17	`$a += 4`	**10**	7	14	8
18	`while ($a <= 10)`	True			
19	`$b = $a + 1`	10	**11**	14	8
20	`$c = $b * 2`	10	11	**22**	8
21	`$d = $c - $b + 1`	10	11	22	**12**
22	`$d == 4`	False			
23	`$d == 5`	False			
24	`$d == 8`	False			
25	`echo $a, $b, $d`	10 11 12 is displayed			
26	`$a += 4`	**14**	11	22	12
27	`while ($a <= 10)`	False			

4. Solution

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

18	`$c = $d`	0	2	**1**	1	1
19	`while ($b < 2)`	False				

5. Solution

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

6. Solution

```
<?php
  $n = trim(fgets(STDIN));
  $sum = 0;

  $i = 1;
  while ($i <= $n) {
    $a = trim(fgets(STDIN));
    $sum = $sum + $a;
    $i++;
  }

  echo $sum;
  if ($n > 0) {
    echo $sum / $n;
  }
?>
```

7. Solution

```
<?php
  $n = trim(fgets(STDIN));
  $p = 1;

  $i = 1;
  while ($i <= $n) {
    $a = trim(fgets(STDIN));
    if ($a % 2 == 0) {
      $p = $p * $a;
    }
    $i++;
  }
  echo $p;
```

```
?>
```

8. Solution

```
<?php
  $sum = 0;

  $i = 1;
  while ($i <= 100) {
    $a = trim(fgets(STDIN));
    if ($a % 10 == 0) {
      $sum = $sum + $a;
    }
    $i++;
  }
  echo $sum;
?>
```

9. Solution

```
<?php
  $sum = 0;

  $i = 1;
  while ($i <= 20) {
    $a = trim(fgets(STDIN));
    if ($a >= 100 && $a <= 999) {
      $sum = $sum + $a;
    }
    $i++;
  }
  echo $sum;
?>
```

10. Solution

```
<?php
  $p = 1;

  $a = trim(fgets(STDIN));
  while ($a != 0) {
    $p = $p * $a;
    $a = trim(fgets(STDIN));
  }
  echo $p;
?>
```

Step	Statement	$a	$p
1	`$p = 1`	?	**1**
2	`$a = trim(fgets(STDIN))`	**3**	1
3	`while ($a != 0)`	True	
4	`$p = $p * $a`	3	**3**
5	`$a = trim(fgets(STDIN))`	**2**	3
6	`while ($a != 0)`	True	
7	`$p = $p * $a`	2	**6**
8	`$a = trim(fgets(STDIN))`	**9**	6
9	`while ($a != 0)`	True	
10	`$p = $p * $a`	9	**54**
11	`$a = trim(fgets(STDIN))`	**0**	54
12	`while ($a != 0)`	False	
13	`echo $p`	54 is displayed	

11. Solution

```
<?php
  $population = 30000;

  $years = 0;
  while ($population <= 100000) {
    $population += $population * 0.03;
    $years++;
  }
  echo $years;
?>
```

Chapter 26

26.2 Answers of Review Questions: True/False

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

26.3 Answers of Review Questions: Multiple Choice

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

26.4 Answers of Review Exercises

1. Solution

```
<?php
  $i = 3;
  do {
    $i--;
  } while ($i > 0);
  echo "The end";
?>
```

2. Solution

Step	Statement	$x	$y
1	`$y = 5`	?	**5**
2	`$x = 38`	**38**	5
3	`$y *= 2`	38	**10**
4	`$x++`	**39**	10
5	`echo $y`	10 is displayed	
6	`while ($y < $x)`	True	
7	`$y *= 2`	39	**20**
8	`$x++`	**40**	20
9	`echo $y`	20 is displayed	
10	`while ($y < $x)`	True	
11	`$y *= 2`	40	**40**
12	`$x++`	**41**	40
13	`echo $y`	40 is displayed	
14	`while ($y < $x)`	True	

15	`$y *= 2`	41	**80**
16	`$x++`	**42**	80
17	`echo $y`	80 is displayed	
18	`while ($y < $x)`	False	

3. Solution

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

4. Solution

Step	Statement	$x	$y
1	`$y = 2`	?	**2**
2	`$x = 0`	**0**	2
3	`$y = pow ($y, 2)`	0	**4**
4	`if ($x < 256)`	True	
5	`$x = $x + $y`	**4**	

6	`echo $x, $y`	4 4 is displayed	
7	`while ($y < 65535)`	True	
8	`$y = pow ($y, 2)`	4	**16**
9	`if ($x < 256)`	True	
10	`$x = $x + $y`	**20**	16
11	`echo $x, $y`	20 16 is displayed	
12	`while ($y < 65535)`	True	
13	`$y = pow ($y, 2)`	20	**256**
14	`if ($x < 256)`	True	
15	`$x = $x + $y`	**276**	256
16	`echo $x, $y`	276 256 is displayed	
17	`while ($y < 65535)`	True	
18	`$y = pow ($y, 2)`	276	**65536**
19	`if ($x < 256)`	False	
20	`echo $x, $y`	276 65536 is displayed	
21	`while ($y < 65535)`	False	

5. Solution

Step	Statement	$a	$b	$c	$d	$x
1	`$a = 2`	**2**	?	?	?	?
2	`$b = 4`	2	**4**	?	?	?
3	`$c = 0`	2	4	**0**	?	?
4	`$d = 0`	2	4	0	**0**	?
5	`$x = $a + $b`	2	4	0	0	**6**
6	`if ($x % 2 != 0)`	False				
7	`elseif ($d % 2 == 0)`	True				
8	`$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	`elseif ($d % 2 == 0)`	False				
21	`$c = $c + 3`	5	5	**8**	5	10
22	`$a = $b`	**5**	5	8	5	10
23	`$b = $d`	5	**5**	8	5	10
24	`$x = $a + $b`	5	5	8	5	**10**
25	`$c = $c + 3`	5	5	**11**	5	10
26	`$a = $b`	**5**	5	11	5	10
27	`$b = $d`	5	**5**	11	5	10
28	`while ($c < 11)`	False				

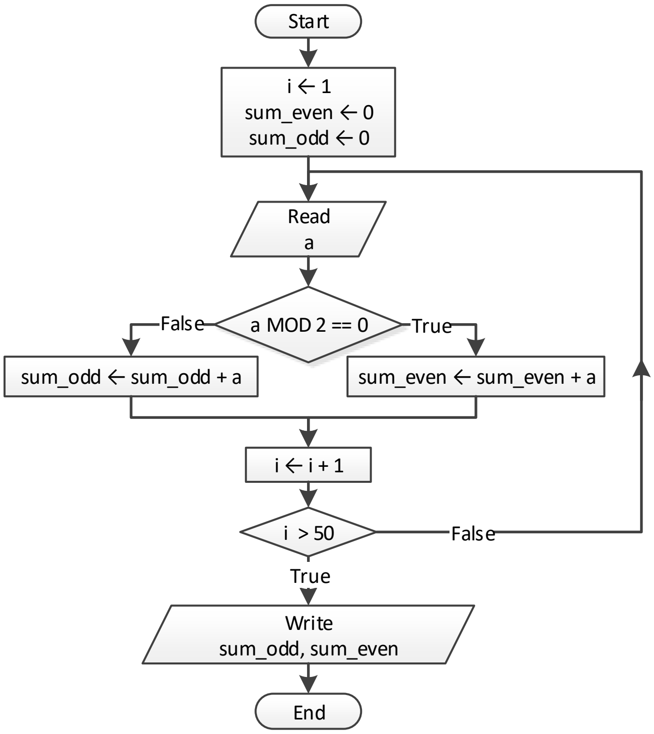
6. Solution

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

7. Solution

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

8. Solution



```
<?php
  $i = 1;
  $sum_even = 0;
  $sum_odd = 0;
  do {
    $a = trim(fgets(STDIN));
    if ($a % 2 == 0) {
      $sum_even += $a;
    }
    else {
      $sum_odd += $a;
    }
    $i++;
  } while ($i <= 50);
  echo $sum_even, $sum_odd;
?>
```

9. Solution

```
<?php
  $n = trim(fgets(STDIN));
  $i = 1;
  $p = 1;
  do {
    $a = trim(fgets(STDIN));
    if ($a < 0) {
      $p *= $a;
    }
    $i++;
  } while ($i <= $n);
  echo abs($p);
?>
```

10. Solution

```
<?php
  $i = 1;
  $p = 1;
  do {
    echo "Enter an integer: ";
    $a = trim(fgets(STDIN));
    if ($a >= 500 && $a <= 599) {
      $p *= $a;
    }
    $i++;
  } while ($i <= 5);
  echo $p;
?>
```

11. Solution

```
<?php
  $sum = 0;

  $a = trim(fgets(STDIN));
  if ($a > 0) {
    do {
      $sum = $sum + $a;
      $a = trim(fgets(STDIN));
    } while ($a > 0);
  }
  echo $sum;
?>
```

Step	Statement	$a	$sum
1	`$sum = 0`	?	**0**
2	`$a = trim(fgets(STDIN))`	**5**	0
3	`if ($a > 0)`	True	
4	`$sum = $sum + $a`	5	**5**
5	`$a = trim(fgets(STDIN))`	**2**	5
6	`while ($a > 0)`	True	

7	`$sum = $sum + $a`	2	**7**
8	`$a = trim(fgets(STDIN))`	**3**	7
9	`while ($a > 0)`	True	
10	`$sum = $sum + $a`	3	**10**
11	`$a = trim(fgets(STDIN))`	**0**	10
12	`while ($a > 0)`	False	

12. Solution

```
<?php
  $population = 50000;

  $years = 0;
  while ($population >= 20000) {
    $population -= $population * 0.10;
    $years++;
  }
  echo $years;
?>
```

Chapter 27

27.3 Answers of Review Questions: True/False

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

27.4 Answers of Review Questions: Multiple Choice

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

27.5 Answers of Review Exercises

1. Solution

Step	Statement	$a	$b	$j
1	`$a = 0`	**0**	?	?
2	`$b = 0`	0	**0**	?
3	`$j = 0`	0	0	**0**
4	`$j <= 8`	True		
5	`if ($j < 5)`	True		
6	`$b++`	0	**1**	0
7	`$j += 2`	0	1	**2**
8	`$j <= 8`	True		
9	`if ($j < 5)`	True		
10	`$b++`	0	**2**	2
11	`$j += 2`	0	2	**4**
12	`$j <= 8`	True		
13	`if ($j < 5)`	True		
14	`$b++`	0	**3**	4
15	`$j += 2`	0	3	**6**
16	`$j <= 8`	True		
17	`if ($j < 5)`	False		

18	$a += $j - 1	**5**	3	6
19	$j += 2	5	3	**8**
20	$j <= 8	True		
21	if ($j < 5)	False		
22	$a += $j - 1	**12**	3	8
23	$j += 2	12	3	**10**
24	$j <= 8	False		
25	echo $a, $b	12 3 is displayed		

2. *Solution*

For input value of 10

Step	Statement	$a	$b	$j
1	$a = trim(fgets(STDIN))	**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	echo $b	24 is displayed		

For input value of 21

Step	Statement	$a	$b	$j
1	$a = trim(fgets(STDIN))	**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	`echo $b`	1 is displayed		

3. *Solution*

For input value of 12

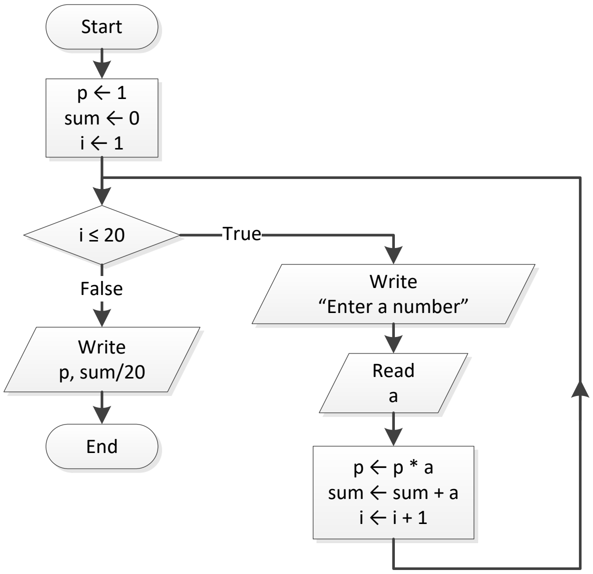
Step	Statement	$a	$x	$y	$j
1	`$a = trim(fgets(STDIN))`	**12**	?	?	?
2	`$j = 2`	12	?	?	**2**
3	`$j <= $a - 1`	True			
4	`$x = $j * 3 + 3`	12	**9**	?	2
5	`$y = $j * 2 + 10`	12	9	**14**	2
6	`if ($y - $x > 0 || $x > 30)`	True			
7	`$y *= 2`	12	9	**28**	2
8	`$x += 4`	12	**13**	28	2
9	`echo $x, $y`	13 28 is displayed			
10	`$j += 3`	12	13	28	**5**
11	`$j <= $a - 1`	True			
12	`$x = $j * 3 + 3`	12	**18**	28	5
13	`$y = $j * 2 + 10`	12	18	**20**	5
14	`if ($y - $x > 0 || $x > 30)`	True			
15	`$y *= 2`	12	18	**40**	5
16	`$x += 4`	12	**22**	40	5
17	`echo $x, $y`	22 40 is displayed			
18	`$j += 3`	12	22	40	**8**
19	`$j <= $a - 1`	True			
20	`$x = $j * 3 + 3`	12	**27**	40	8

21	`$y = $j * 2 + 10`	12	27	**26**	8
22	`if ($y - $x > 0 || $x > 30)`	False			
23	`$x += 4`	12	**31**	26	8
24	`echo $x, $y`	31 26 is displayed			
25	`$j += 3`	12	31	26	**11**
26	`$j <= $a - 1`	True			
27	`$x = $j * 3 + 3`	12	**36**	26	11
28	`$y = $j * 2 + 10`	12	36	**32**	11
29	`if ($y - $x > 0 || $x > 30)`	True			
30	`$y *= 2`	12	36	**64**	11
31	`$x += 4`	12	**40**	64	11
32	`echo $x, $y`	40 64 is displayed			
33	`$j += 3`	12	40	64	**14**
34	`$j <= $a - 1`	False			

4. Solution

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

5. Solution

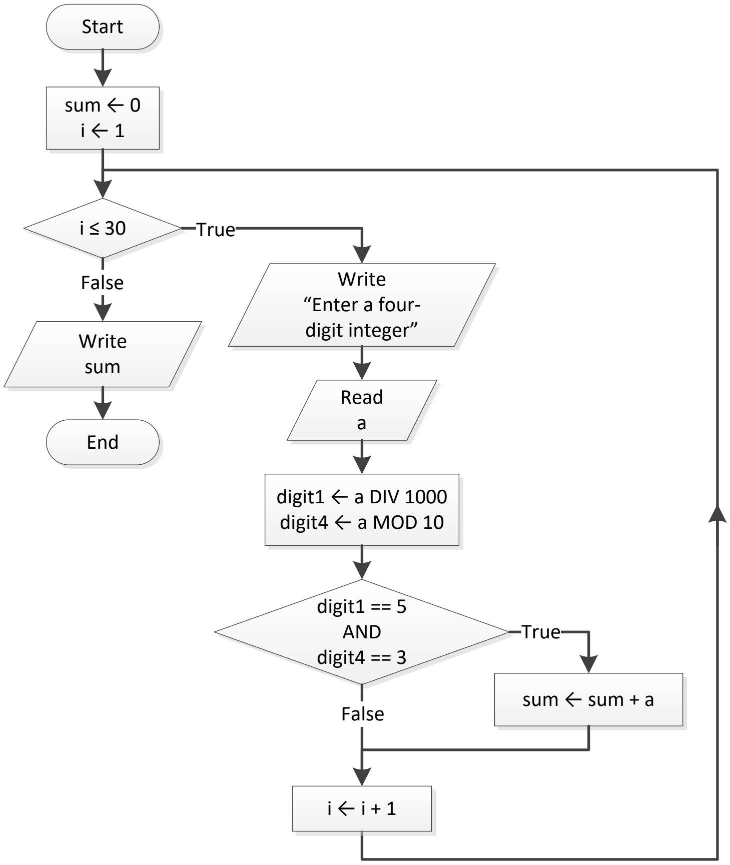


```
<?php
  $p = 1;
  $sum = 0;
  for ($i = 1 ; $i <= 20; $i++) {
    echo "Enter a number: ";
    $a = trim(fgets(STDIN));
    $p = $p * $a;
    $sum = $sum + $a;
  }
  echo $p, $sum / 20;
?>
```

6. Solution

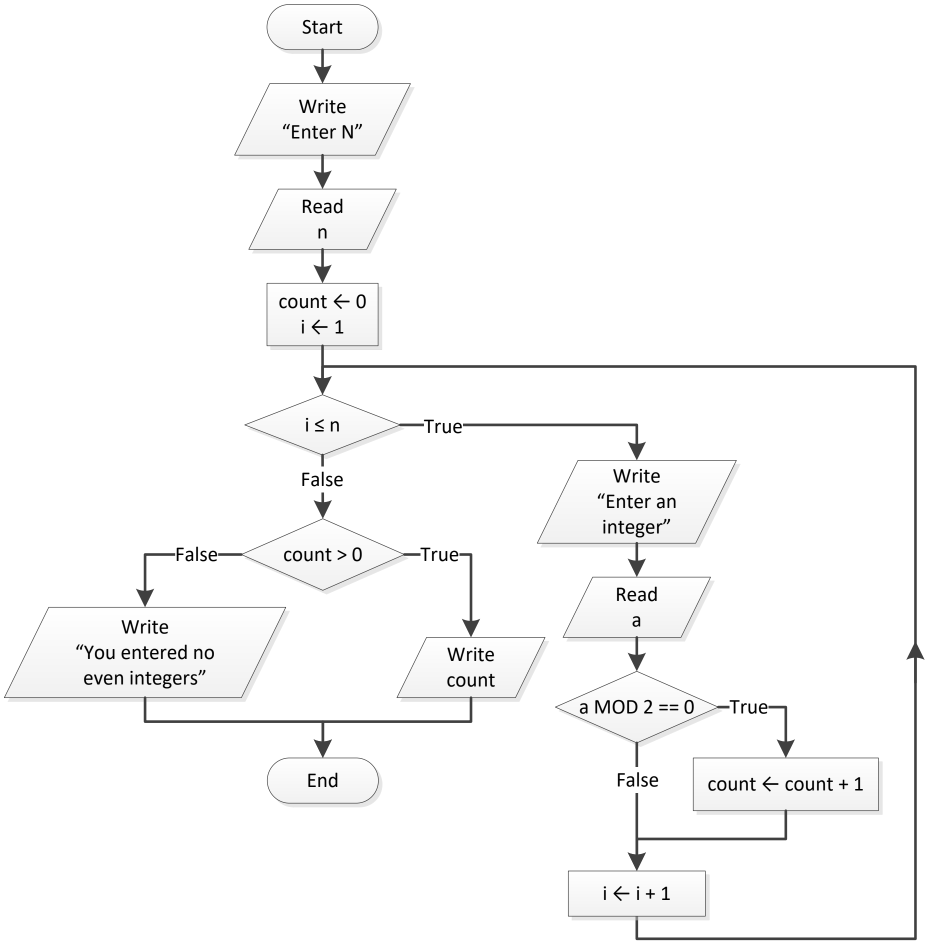
```
<?php
  echo "Enter degrees: ";
  $deg = trim(fgets(STDIN));
  for ($i = 0 ; $i <= $deg; $i += 0.5) {
    echo sin($i * pi() /180);
  }
?>
```

7. Solution



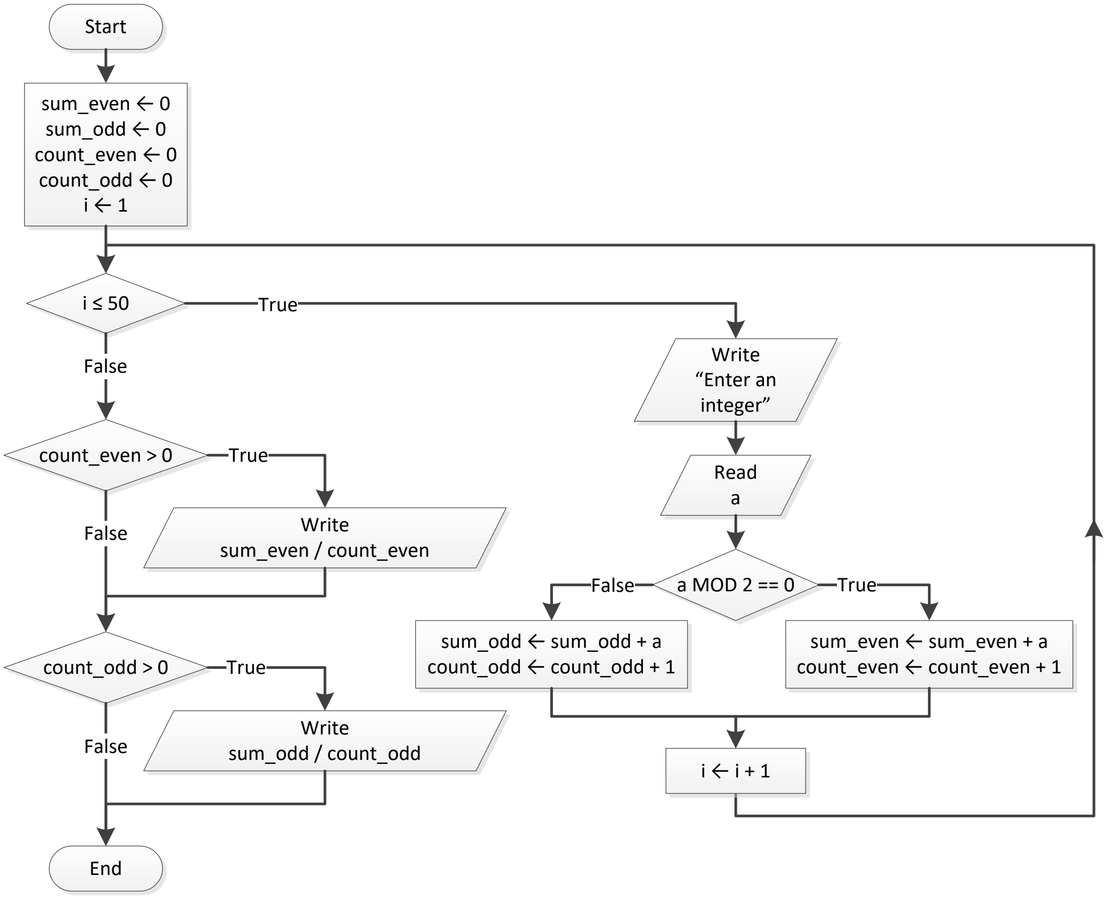
```
<?php
  $sum = 0;
  for ($i = 1; $i <= 30; $i++) {
    echo "Enter a four-digit integer: ";
    $a = trim(fgets(STDIN));
    $digit1 = intval($a / 1000);
    $digit4 = $a % 10;
    if ($digit1 == 5 && $digit4 == 3) {
      $sum += $a;
    }
  }
  echo $sum;
?>
```

8. Solution



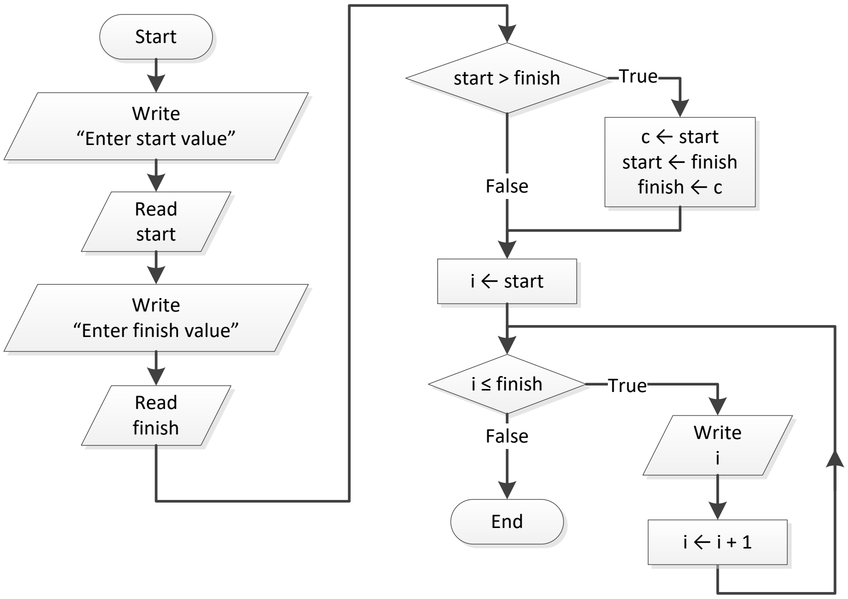
```
<?php
  echo "Enter N: ";
  $n = trim(fgets(STDIN));
  $count = 0;
  for ($i = 0 ; $i <= $n; $i++) {
    echo "Enter an integer: ";
    $a = trim(fgets(STDIN));
    if ($a % 2 == 0) {
      $count++;
    }
  }
  if ($count > 0) {
    echo $count;
  }
  else {
    echo "You entered no even integers";
  }
?>
```

9. Solution



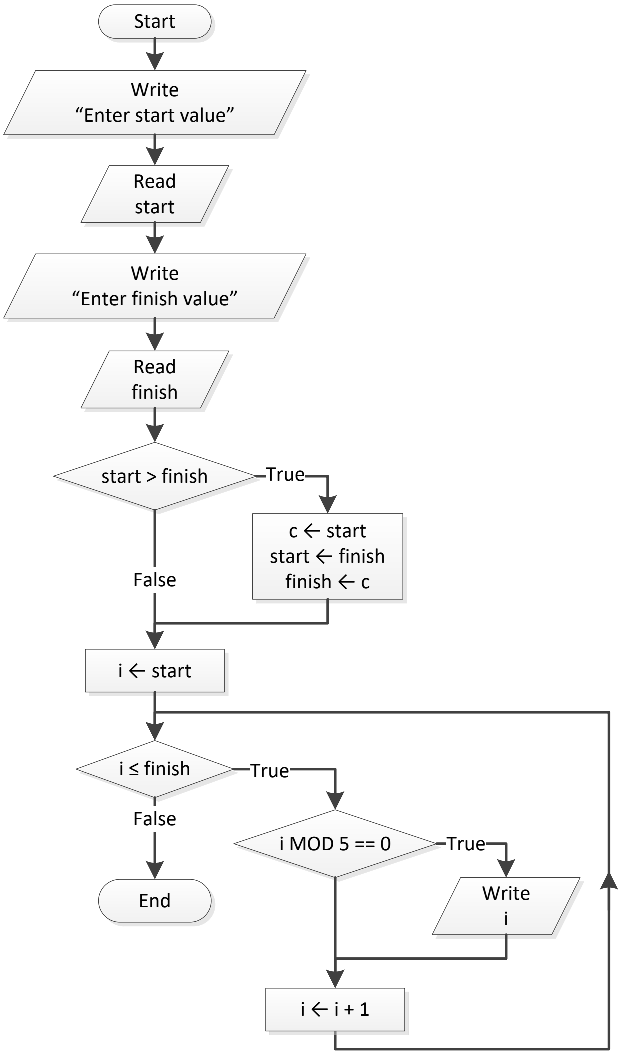
```
<?php
  $sum_even = 0;
  $sum_odd = 0;
  $count_even = 0;
  $count_odd = 0;
  for ($i = 1; $i <= 50; $i++) {
    echo "Enter an integer: ";
    $a = trim(fgets(STDIN));
    if ($a % 2 == 0) {
      $sum_even += $a;
      $count_even++;
    }
    else {
      $sum_odd += $a;
      $count_odd++;
    }
  }
  if ($count_even > 0) {
    echo $sum_even / $count_even;
  }
  if ($count_odd > 0) {
    echo $sum_odd / $count_odd;
  }
?>
```

10. Solution



```
<?php
  echo "Enter start value: ";
  $start = trim(fgets(STDIN));
  echo "Enter finish value: ";
  $finish = trim(fgets(STDIN));

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

  for ($i = $start; $i <= $finish; $i++) {
    echo $i;
  }
?>
```

11. Solution



```
<?php
  echo "Enter start value: ";
  $start = trim(fgets(STDIN));
  echo "Enter finish value: ";
  $finish = trim(fgets(STDIN));

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

  for ($i = $start; $i <= $finish; $i++) {
    if ($i % 5 == 0) {
      echo $i;
    }
  }
?>
```

12. Solution

First Approach

```
<?php
  echo "Enter a value for base: ";
  $base = trim(fgets(STDIN));
  echo "Enter an integer for exponent: ";
  $exp = trim(fgets(STDIN));

  $p = 1;
  if ($exp >= 0) {
    for ($i = 1; $i <= $exp; $i++) {
      $p *= $base;
    }
  }
  else {
    for ($i = 1; $i <= -$exp; $i++) {
      $p *= 1 / $base;
    }
  }
  echo $p;
?>
```

Second Approach

```
<?php
  echo "Enter a value for base: ";
  $base = trim(fgets(STDIN));
  echo "Enter an integer for exponent: ";
  $exp = trim(fgets(STDIN));

  $p = 1;
  for ($i = 1; $i <= abs($exp); $i++) {
    $p *= $base;
  }
  if ($exp < 0) {
    $p = 1 / $p;
  }
  echo $p;
?>
```

13. Solution

```
<?php
  echo "Enter a message: ";
  $msg = trim(fgets(STDIN));

  $characters = strlen($msg);
  $count = 0;
  for ($i = 0; $i <= $characters - 1; $i++) {
    if ($msg[$i] == " ") {
      $count++;
    }
  }
  $words = $count + 1;
  echo "The message entered contains ",$words, " words";
?>
```

14. Solution

```
<?php
  echo "Enter a message: ";
  $msg = trim(fgets(STDIN));

  $characters = strlen($msg);
  $count = 0;
  for ($i = 0; $i <= $characters - 1; $i++) {
    if ($msg[$i] == " ") {
      $count++;
    }
  }
  $words = $count + 1;
  echo "The average number of letters in each word is ", ($characters - $count) / $words;
?>
```

Chapter 28

28.3 Answers of Review Questions: True/False

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

28.4 Answers of Review Questions: Multiple Choice

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

28.5 Answers of Review Exercises

1. Solution

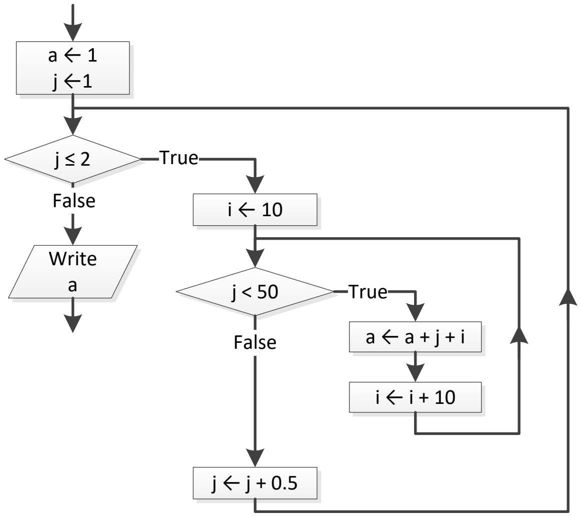
i. 10

ii. 4.5

iii. -7

iv. 138

2. Solution



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

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

3. Solution

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

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

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

4. Solution

For input value of “NO”

Step	Statement	$s	$y	$i	$ans
1	`$s = 1`	**1**	?	?	?
2	`$y = 25`	1	**25**	?	?
3	`$i = 1`	1	25	**1**	?
4	`$i <= 3`	True			
5	`$s = $s + $y`	**26**	25	1	?
6	`$y -= 5`	26	**20**	1	?
7	`$i++`	26	20	**2**	?
8	`$i <= 3`	True			
9	`$s = $s + $y`	**46**	20	2	?
10	`$y -= 5`	46	**15**	2	?
11	`$i++`	46	15	**3**	?
12	`$i <= 3`	True			
13	`$s = $s + $y`	**61**	15	3	?
14	`$y -= 5`	61	**10**	3	?
15	`$i++`	61	10	**4**	?
16	`$i <= 3`	False			
17	`$ans = trim(fgets(STDIN))`	61	10	4	**“NO”**
18	`while ($ans == "YES")`	False			
19	`echo $s`	61 is displayed			

For input values of “YES”, “NO”

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

12	`$i <= 3`	True			
13	`$s = $s + $y`	**61**	15	3	?
14	`$y -= 5`	61	**10**	3	?
15	`$i++`	61	10	**4**	?
16	`$i <= 3`	False			
17	`$ans = trim(fgets(STDIN))`	61	10	4	**“YES”**
18	`while ($ans == "YES")`	True			
19	`$i = 1`	61	10	**1**	“YES”
20	`$i <= 3`	True			
21	`$s = $s + $y`	**71**	10	1	“YES”
22	`$y -= 5`	71	**5**	1	“YES”
23	`$i++`	71	5	**2**	“YES”
24	`$i <= 3`	True			
25	`$s = $s + $y`	**76**	5	2	“YES”
26	`$y -= 5`	76	**0**	2	“YES”
27	`$i++`	76	0	**3**	“YES”
28	`$i <= 3`	True			
29	`$s = $s + $y`	**76**	0	3	“YES”
30	`$y -= 5`	76	**-5**	3	“YES”
31	`$i++`	76	-5	**4**	“YES”
32	`$i <= 3`	False			
33	`$ans = trim(fgets(STDIN))`	76	-5	4	**“NO”**
34	`while ($ans == "YES")`	False			
35	`echo $s`	76 is displayed			

For input values of “YES”, “YES”, “NO”

Step	Statement	$s	$y	$i	$ans
1	`$s = 1`	**1**	?	?	?
2	`$y = 25`	1	**25**	?	?
3	`$i = 1`	1	25	**1**	?
4	`$i <= 3`	True			
5	`$s = $s + $y`	**26**	25	1	?
6	`$y -= 5`	26	**20**	1	?
7	`$i++`	26	20	**2**	?
8	`$i <= 3`	True			
9	`$s = $s + $y`	**46**	20	2	?
10	`$y -= 5`	46	**15**	2	?

11	`$i++`	46	15	**3**	?
12	`$i <= 3`	True			
13	`$s = $s + $y`	**61**	15	3	?
14	`$y -= 5`	61	**10**	3	?
15	`$i++`	61	10	**4**	?
16	`$i <= 3`	False			
17	`$ans = trim(fgets(STDIN))`	61	10	4	**“YES”**
18	`while ($ans == "YES")`	True			
19	`$i = 1`	61	10	**1**	“YES”
20	`$i <= 3`	True			
21	`$s = $s + $y`	**71**	10	1	“YES”
22	`$y -= 5`	71	**5**	1	“YES”
23	`$i++`	71	5	**2**	“YES”
24	`$i <= 3`	True			
25	`$s = $s + $y`	**76**	5	2	“YES”
26	`$y -= 5`	76	**0**	2	“YES”
27	`$i++`	76	0	**3**	“YES”
28	`$i <= 3`	True			
29	`$s = $s + $y`	**76**	0	3	“YES”
30	`$y -= 5`	76	**-5**	3	“YES”
31	`$i++`	76	-5	**4**	“YES”
32	`$i <= 3`	False			
33	`$ans = trim(fgets(STDIN))`	76	-5	4	**“YES”**
34	`while ($ans == "YES")`	True			
35	`$i = 1`	76	-5	**1**	“YES”
36	`$i <= 3`	True			
37	`$s = $s + $y`	**71**	-5	1	“YES”
38	`$y -= 5`	71	**-10**	1	“YES”
39	`$i++`	71	-10	**2**	“YES”
40	`$i <= 3`	True			
41	`$s = $s + $y`	**61**	-10	2	“YES”
42	`$y -= 5`	61	**-15**	2	“YES”
43	`$i++`	61	-15	**3**	“YES”
44	`$i <= 3`	True			
45	`$s = $s + $y`	**46**	-15	3	“YES”
46	`$y -= 5`	46	**-20**	3	“YES”

47	`$i++`	46	-20	**4**	"YES"
48	`$i <= 3`	False			
49	`$ans = trim(fgets(STDIN))`	46	-20	4	**"NO"**
50	`while ($ans == "YES")`	False			
51	`echo $s`	46 is displayed			

5. Solution

```
<?php
  for ($hour = 0; $hour <= 23; $hour++) {
    for ($minutes = 0; $minutes <= 59; $minutes++) {
      echo $hour, "\t", $minutes, "\n";
    }
  }
?>
```

6. Solution

```
<?php
  for ($i = 5; $i >= 1; $i--) {
    for ($j = 1; $j <= $i; $j++) {
      echo $i, " ";
    }
    echo "\n";
  }
?>
```

7. Solution

```
<?php
  for ($i = 0; $i <= 5; $i++) {
    for ($j = 0; $j <= $i; $j++) {
      echo $j, " ";
    }
    echo "\n";
  }
?>
```

8. Solution

```
<?php
  for ($i = 1; $i <= 4; $i++) {
    for ($j = 1; $j <= 10; $j++) {
      echo "* ";
    }
    echo "\n";
  }
?>
```

9. Solution

```
<?php
  echo "Enter an integer between 3 and 20: ";
  $n = trim(fgets(STDIN));

  for ($i = 1; $i <= $n; $i++) {
    for ($j = 1; $j <= $n; $j++) {
      echo "* ";
    }
    echo "\n";
  }
?>
```

10. Solution

```
<?php
  echo "Enter an integer between 3 and 20: ";
  $n = trim(fgets(STDIN));

  for ($j = 1; $j <= $n; $j++) {
    echo "* ";
  }
  echo "\n";

  for ($i = 1; $i <= $n - 2; $i++) {
    echo "* ";
    for ($j = 1; $j <= $n - 2; $j++) {
      echo "  ";
    }
    echo "* \n";
  }

  for ($j = 1; $j <= $n; $j++) {
    echo "* ";
  }
?>
```

11. Solution

```
<?php
  for ($i = 1; $i <= 5; $i++) {
    for ($j = 1; $j <= $i; $j++) {
      echo "* ";
    }
    echo "\n";
  }

  for ($i = 4; $i >= 1; $i--) {
    for ($j = 1; $j <= $i; $j++) {
      echo "* ";
    }
    echo "\n";
  }
?>
```

Chapter 29

29.14 Answers of Review Questions: True/False

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

29.15 Answers of Review Questions: Multiple Choice

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

29.16 Answers of Review Exercises

1. Solution

```
$s = 0;
for ($i = 1; $i <= 100; $i++) {
  $number = trim(fgets(STDIN));
  $s = $s + $number;
}
$average = $s / 100;
echo $average;
```

2. Solution

```
<?php
  $s = 0;

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

  for ($i = 1; $i <= 100; $i++) {
    $s += $i / $denom;
  }
  echo $s;
?>
```

3. Solution

```
$s = 10;
$i = 1;
while ($i <= 10) {
  $s += sqrt($i);
  $i++;
}
echo $s;
```

4. Solution

```
$start = trim(fgets(STDIN));
$end = trim(fgets(STDIN));
$i = $start;
while ($i <= $end) {
  echo $i;
  $i++;
}
```

5. Solution

```
$s = 0;
for ($i = 100; $i >= 5; $i -= 5) {
  $s = $s + sqrt($i);
}
echo $s;
```

6. Solution

```
$s = 0;
$y = 0;
for ($i = 1; $i <= 10;  $i++) {
  $s = $s + sqrt($y + $i + 1);
  $y = $y + ($i + 1) * 2;
}
echo $s;
```

7. Solution

```
$y = 0;
for ($i = 1; $i <= 9; $i += 2) {
  $a = trim(fgets(STDIN));
  $a += $i;
  $y = $y + pow($a + $i + 2, 3);
}
echo $y;
```

8. Solution

This conversion should not be carried out.

9. Solution

```
$s = 0;
$a = trim(fgets(STDIN));
$s += $a;
$a = trim(fgets(STDIN));
while ($a <= $s) {
  $s += $a;
  $a = trim(fgets(STDIN));
}
echo $s;
```

10. Solution

```
$a = 100;
$count = 0;
echo $a;
$b = trim(fgets(STDIN));
$count++;
$a -= sqrt($b);
while ($a >= 0) {
  echo $a;
  $b = trim(fgets(STDIN));
  $count++;
  $a -= sqrt($b);
}
echo $count;
```

11. Solution

```
$a = trim(fgets(STDIN));
$b = trim(fgets(STDIN));
if ($b <= 1000) {
  do {
    $a += 2;
    $b = $b * $a;
    echo $b;
  } while ($b <= 1000);
}
```

12. Solution

```
$s = 0;
$a = trim(fgets(STDIN));
if ($a != -99) {
  do {
    $s = $s + pow($a, 2);
    $a = trim(fgets(STDIN));
  } while ($a != -99);
}
echo $s;
```

13. Solution

```
$x = 0;
$y = -10;
do {
  $x = $x + pow(2, $y);
  $y = $y + 1;
} while ($y < 10);
echo $x;
```

14. Solution

```
$start = trim(fgets(STDIN));
$x = 1;
$i = $start;
while ($i <= $start * 2) {
  $x = pow($x, 1.1) + $i;
  $i++;
}
echo $x;
```

15. Solution

```
$x = 42;
$i = 1;
while ($i <= 100) {
  $x = sqrt($x) + $i;
  echo $x, "\n";
  $i++;
}
```

16. Solution

```
<?php
  $p = 1;
  $a = trim(fgets(STDIN));
  $i = $a;
  $p = $p * pow($i, 2);
  $i = $i + 5;
  $p = $p + $i;
  for ($i = $a + 5; $i <= 19; $i += 5) {
    $p = $p * pow($i, 2);
    $p = $p + $i + 5;
  }
  echo $p;
?>
```

17. Solution

```
<?php
  $start = trim(fgets(STDIN));
  $end = trim(fgets(STDIN));
  $x = 1000;

  for ($i = $start; $i <= $end; $i+=2) {
    $x = sqrt($x);
  }
  echo $x;
?>
```

18. Solution

```
<?php
  for ($i = 1; $i <= 4; $i++) {
    for ($j = 1; $j <= 4; $j++) {
      echo $i, " x ", $j, " = ", $i * $j, "\n";
    }
  }
?>
```

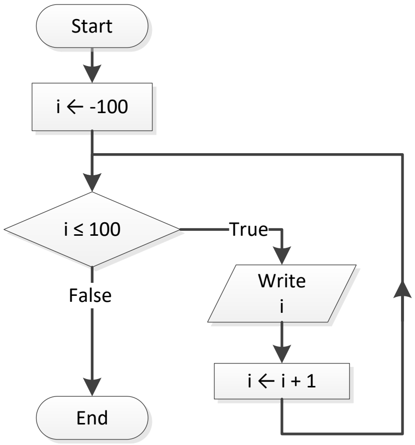
19. Solution

```
<?php
  echo "\t|\t";
  for ($i = 1; $i <= 12; $i++) {
    echo $i, "\t";
  }
  echo "\n";

  for ($i = 1; $i <= 12; $i++) {
    echo "---------";
  }
  echo "\n";
  for ($i = 1; $i <= 12; $i++) {
    echo $i, "\t|\t";
    for ($j = 1; $j <= 12; $j++) {
      echo $i * $j, "\t";
    }
    echo "\n";
  }
?>
```

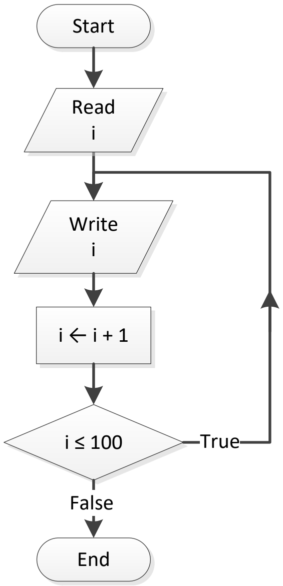
Chapter 30

30.4 Answers of Review Exercises

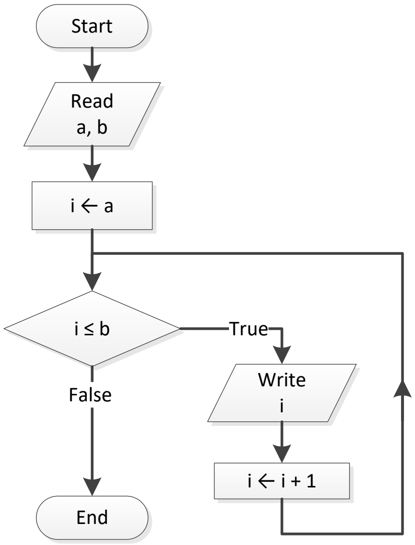
1. Solution



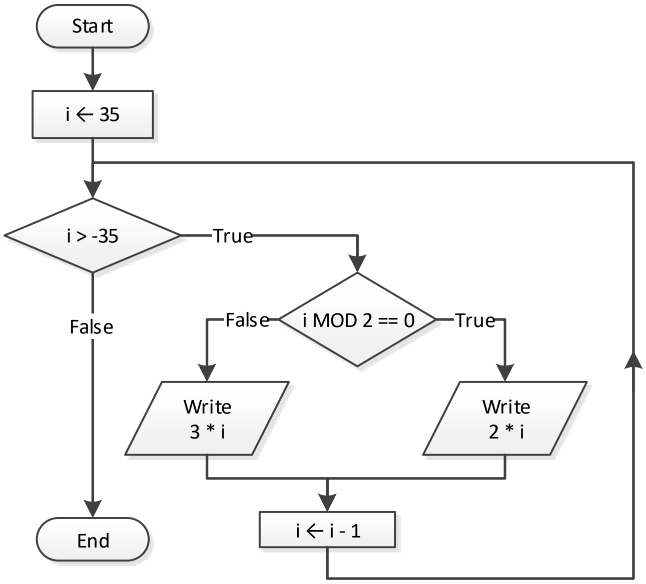
2. Solution



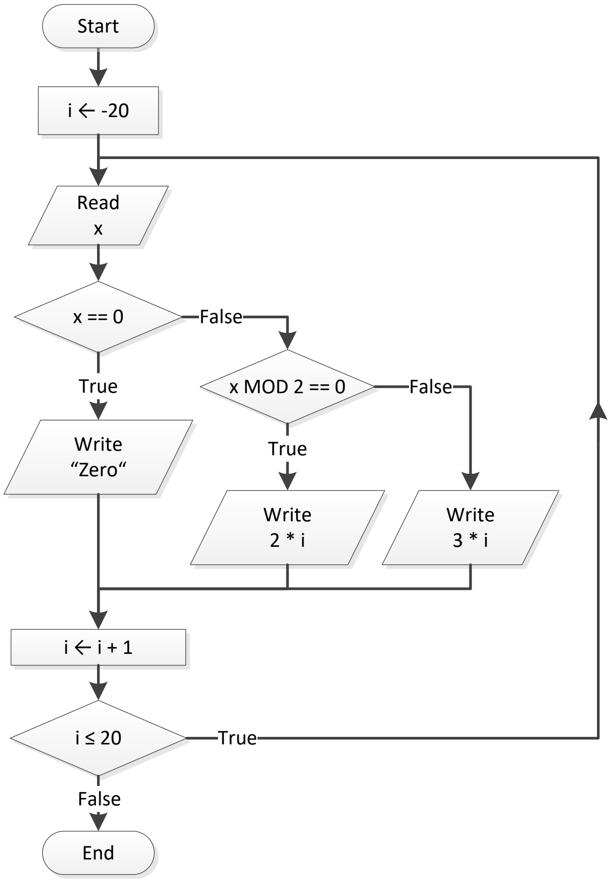
3. Solution



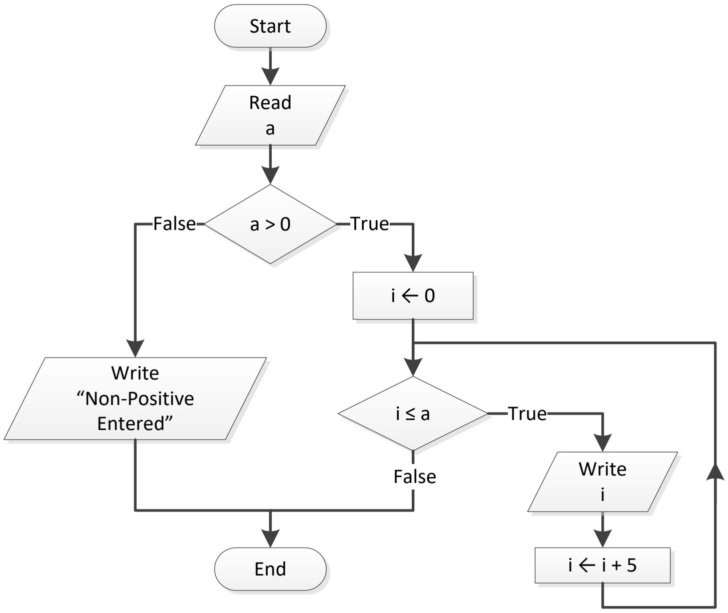
4. Solution



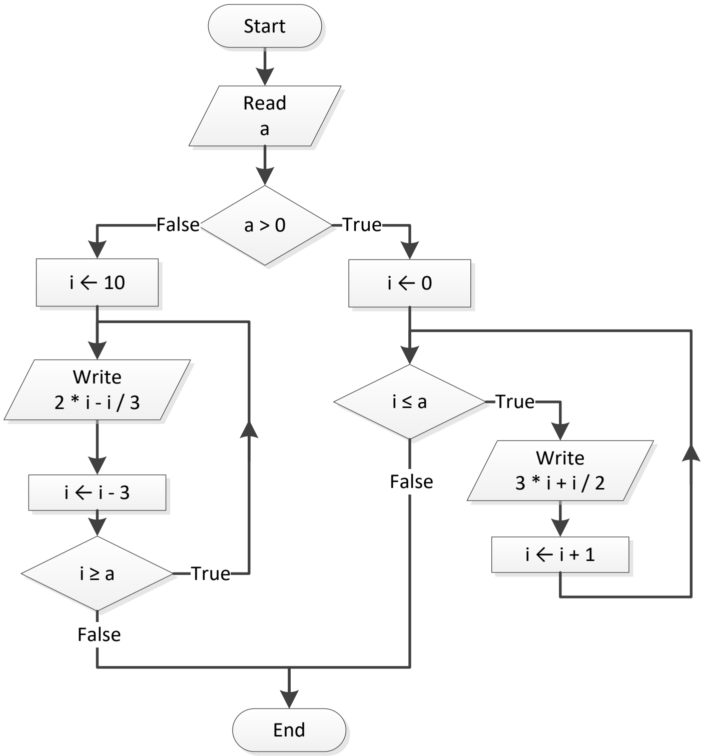
5. Solution



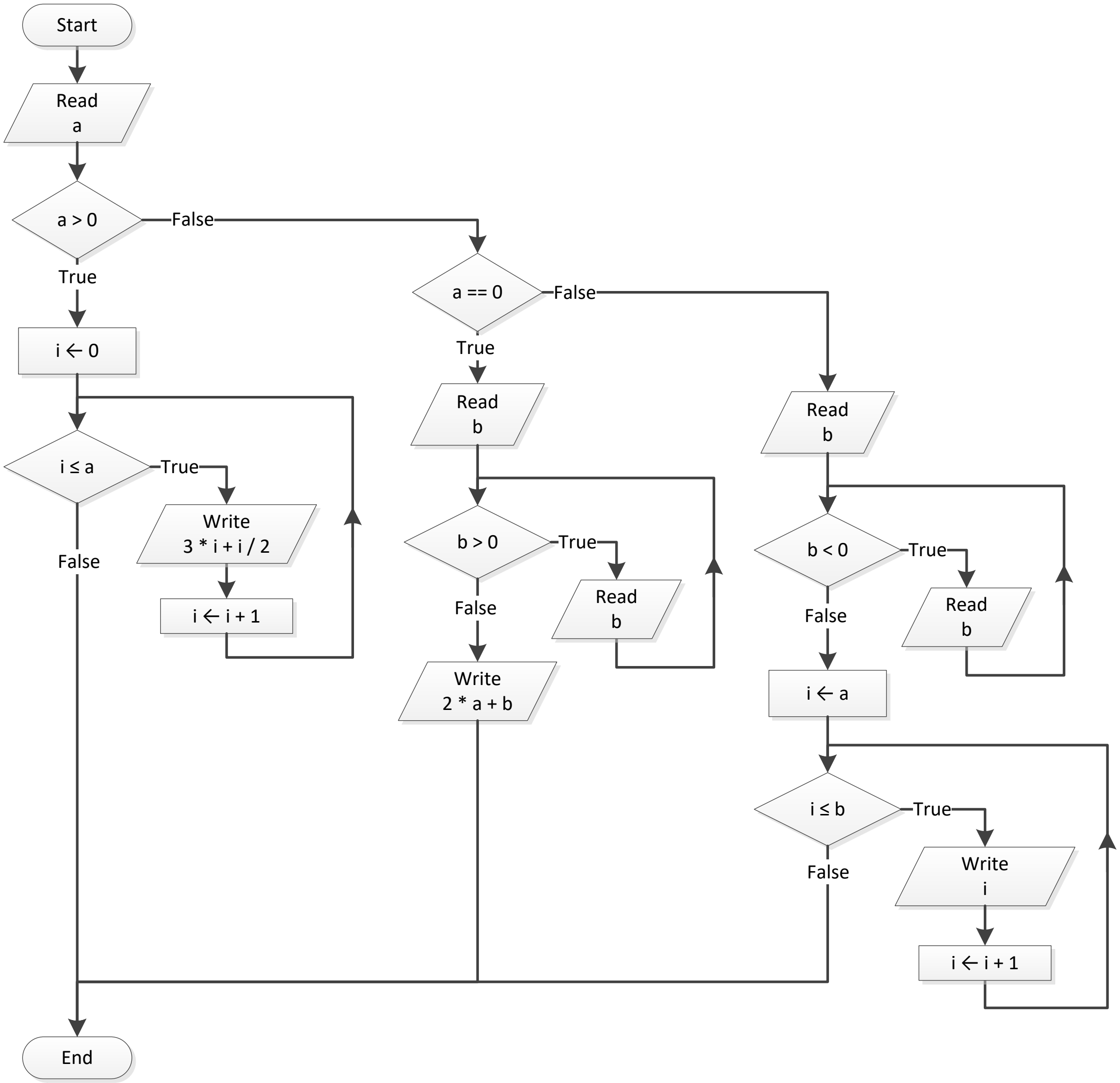
6. Solution



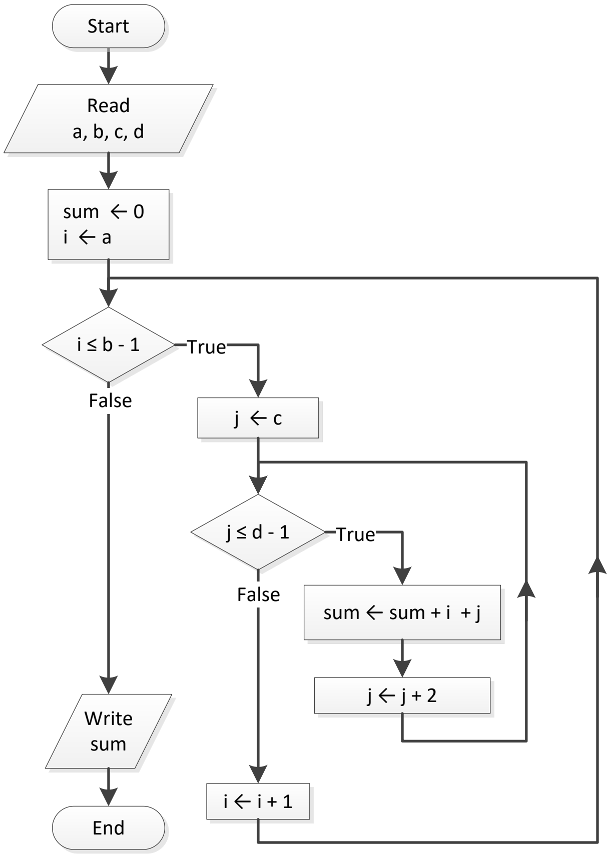
7. Solution


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

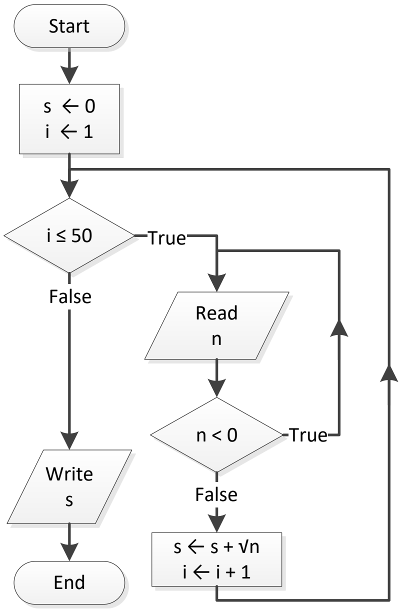
8. Solution


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

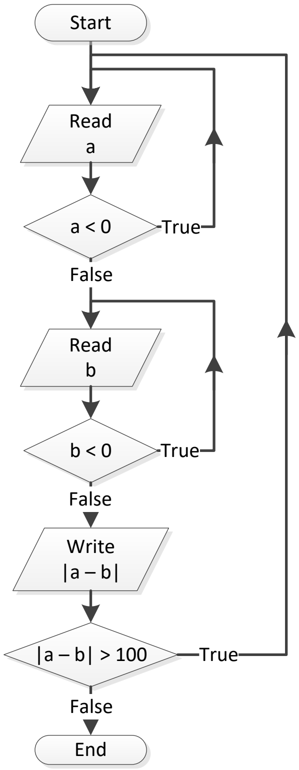
9. Solution



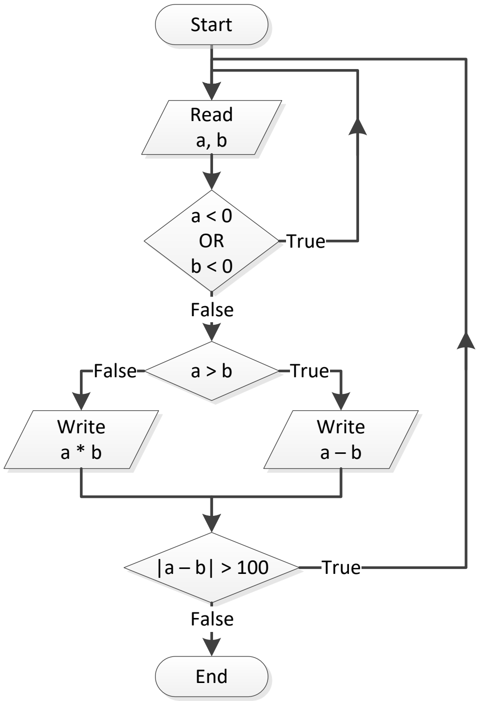
10. Solution



11. Solution



12. Solution



13. Solution

```
$i = 1;
do {
  echo $i;
  $i += 5;
} while ($i <= 500);
echo "The End";
```

14. Solution

```
<?php
  $i = 0;
  $a = trim(fgets(STDIN));
  do {
    if ($i % 2 != 0) {
      echo $i;
    }
    $i += 5;
  } while ($i < $a);
?>
```

15. Solution

```
<?php
  $a = trim(fgets(STDIN));
  while ($a != -1) {
    do {
      $b = trim(fgets(STDIN));
    } while ($b <= $a);
    for ($i = $a; $i <= $b; $i++) {
      echo $i;
    }
    $a = trim(fgets(STDIN));
  }
?>
```

16. Solution

```
<?php
  $i = 1;
  $S = 0;
  $P = 1;
  $a = 0;
  if ($i < 45) {
    $S += $a;
  }
  else {
    $P *= a;
  }
  $i++;
  while ($i < 90) {
    $a = trim(fgets(STDIN));
    if ($i < 45) {
      $S += $a;
    }
    else {
      $P *= a;
    }
    $i++;
  }
  echo $S, $P;
?>
```

Chapter 31

31.7 Answers of Review Questions: True/False

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

31.8 Answers of Review Exercises

1. Solution

```
<?php
  $s = 0;
  for ($i = 1; $i <= 99; $i += 2) {
    $s += $i;
  }
  echo $s;
?>
```

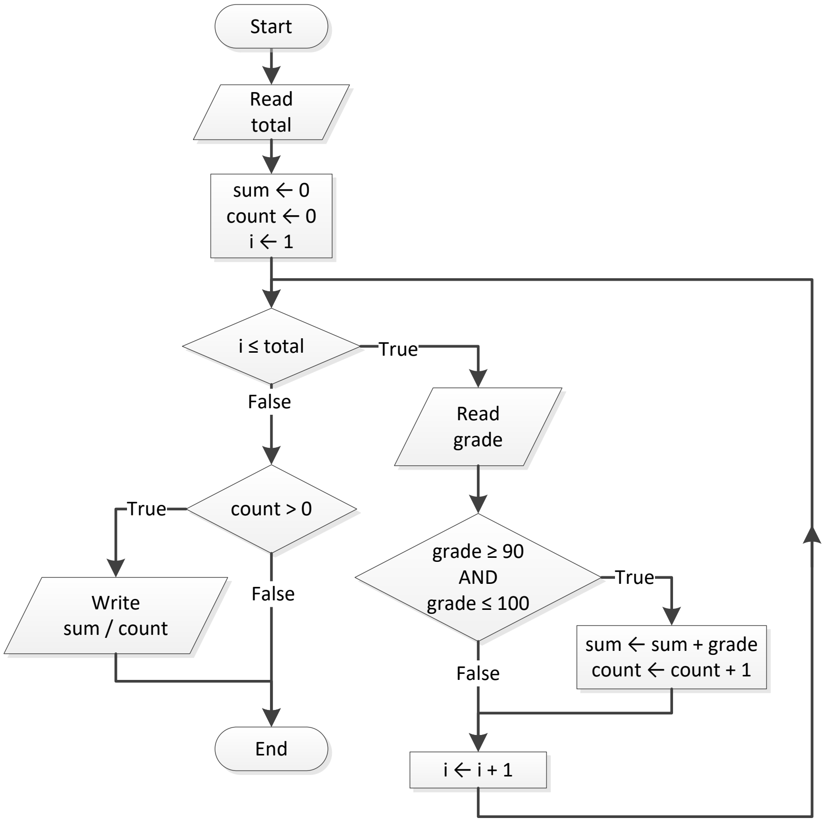
2. Solution

```
<?php
  $n = trim(fgets(STDIN));
  $p = 1;
  for ($i = 2; $i <= 2 * $n; $i += 2) {
    $p *= pow($i, $i - 1);
  }
  echo $p;
?>
```

3. Solution

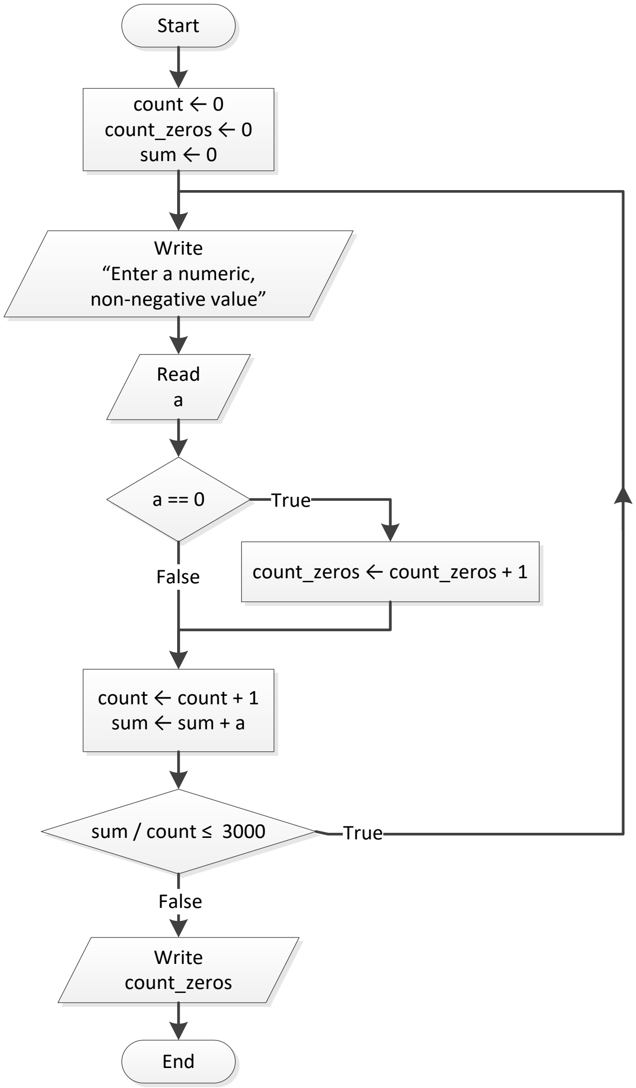
```
<?php
  $s = 0;
  $i = 1;
  $offset = 0;
  while ($i <= 191) {
    $s += $i;
    $offset++;
    $i += $offset;
  }
  echo $s;
?>
```

4. Solution



```php
<?php
  $total = trim(fgets(STDIN));
  $sum = 0;
  $count = 0;
  for ($i = 1; $i <= $total; $i++) {
    $grade = trim(fgets(STDIN));
    if ($grade >= 90 && $grade <= 100) {
      $sum += $grade;
      $count++;
    }
  }
  if ($count > 0) {
    echo $sum / $count;
  }
?>
```

5. Solution



```php
<?php
  $count = 0;
  $count_zeros = 0;
  $sum = 0;
  do {
    echo "Enter a numeric, non-negative value: ";
    $a = trim(fgets(STDIN));
    if ($a == 0) {
      $count_zeros++;
    }
    $count++;
    $sum += $a;
  } while ($sum / $count <= 3000);
  echo $count_zeros;
```

```
?>
```

6. Solution

First Approach

```
<?php
  echo "Enter an integer between 1 and 20: ";
  $a = trim(fgets(STDIN));
  for ($i = 1000; $i <= 9999; $i++) {
    $d4 = $i % 10;
    $r = intval($i / 10);
    $d3 = $r % 10;
    $r = intval($r / 10);
    $d2 = $r % 10;
    $d1 = intval($r / 10);
    if ($d1 + $d2 + $d3 + $d4 < $a) {
      echo $i , "\n";
    }
  }
?>
```

Second Approach

```
<?php
  echo "Enter an integer between 1 and 20: ";
  $a = trim(fgets(STDIN));
  for ($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) {
            echo $d1 * 1000 + $d2 * 100 + $d3 * 10 + $d4 , "\n";
          }
        }
      }
    }
  }
?>
```

7. Solution

First Approach

```
<?php
  for ($i = 1000; $i <= 9999; $i++) {
    $d4 = $i % 10;
    $r = intval($i / 10);
    $d3 = $r % 10;
    $r = intval($r / 10);
    $d2 = $r % 10;
    $d1 = intval($r / 10);
    if ($d1 > $d2 && $d2 == $d3 && $d3 < $d4) {
      echo $i , "\n";
    }
  }
?>
```

Second Approach

```
<?php
  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) {
            echo $d1 * 1000 + $d2 * 100 + $d3 * 10 + $d4 , "\n";
          }
        }
      }
    }
  }
?>
```

8. Solution

```
$x = trim(fgets(STDIN));
while ((is_numeric($x) != true) || ($x != 1) && ($x != 0)) {
  echo "Error!";
  $x = trim(fgets(STDIN));
}
```

9. Solution

```
do {
  $sex = strtoupper(trim(fgets(STDIN)));
} while ($sex != "M" && $sex != "F");
```

10. Solution

```
<?php
  echo "Enter a non-negative number: ";
  $x = trim(fgets(STDIN));
  $count = 1;
  while ($count < 3 && (is_numeric($x) != true || $x < 0)) {
    echo "Error: Invalid number! \n";
    echo "Enter a non-negative number: ";
    $x = trim(fgets(STDIN));
    if (is_numeric($x) != true || $x < 0) {
      $count++;
    }
  }

  if ($count < 3) {
    $y = sqrt($x);
    echo $y;
  }
  else {
    echo "Dude, you are dumb!";
  }
?>
```

11. Solution

```
<?php
  do {
    echo "Enter the length of a radius: ";
    $r = trim(fgets(STDIN));
    while ($r <= 0) {
      echo "Invalid radius. Enter the length of a radius: ";
      $r = trim(fgets(STDIN));
    }

    $area = pi() * pow($r, 2);
    echo "The area is: ", $area, "\n";

    echo "Would you like to repeat? ";
    $answer = trim(fgets(STDIN));
  } while (strtoupper($answer) == "YES");
?>
```

12. Solution

```
<?php
  for ($x = -100; $x <= 100; $x++) {
    for ($y = -100; $y <= 100; $y++) {
      if (5 * $x + 3 * pow($y, 2) == 0) {
        echo $x, ",", $y, "\n";
      }
    }
  }
?>
```

13. Solution

```
<?php
```

```
  for ($x = -10; $x <= 10; $x++) {
    for ($y = -10; $y <= 10; $y++) {
      for ($z = -10; $z <= 10; $z++) {
        if (($x + $y) / 2 + 3 * pow($z, 2) / ($x + 3 * $y + 45) == $x / 3) {
          echo $x, ",", $y, ",", $z, "\n";
        }
      }
    }
  }
?>
```

14. Solution

```
<?php
  $m1 = trim(fgets(STDIN));
  $m2 = trim(fgets(STDIN));
  $m3 = trim(fgets(STDIN));

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

  $m1 = $s;
  $m2 = $m3;

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

  echo $s;
?>
```

15. Solution

```
<?php
  echo "Enter an integer greater than 1: ";
  $a = trim(fgets(STDIN));
  while ($a < 2) {
    echo "Wrong number. Please enter an integer greater than 1: ";
    $a = trim(fgets(STDIN));
  }

  echo "Enter a second integer greater than 1: ";
  $b = trim(fgets(STDIN));
  while ($b < 2) {
    echo "Wrong number. Please enter a second integer greater than 1: ";
    $b = trim(fgets(STDIN));
  }

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

  for ($x = $a; $x <= $b; $x++) {
    $number_of_divisors = 2;
    $i = 2;
    while ($i <= intval($x / 2) && $number_of_divisors == 2) {
      if ($x % $i == 0) {
        $number_of_divisors++;
      }
      $i++;
    }
    if ($number_of_divisors == 2) {
      echo "Number ", $x, " is prime\n";
    }
  }
?>
```

16. Solution

```
<?php
  echo "Enter a four-digit integer: ";
  $a = trim(fgets(STDIN));
  while (is_numeric($a) != true || $a < 1000 || $a > 9999) {
    echo "Wrong number. Please enter a four-digit integer: ";
    $a = trim(fgets(STDIN));
  }

  echo "Enter a second four-digit integer: ";
  $b = trim(fgets(STDIN));
  while (is_numeric($b) != true || $b < 1000 || $b > 9999) {
    echo "Wrong number. Please enter a second four-digit integer: ";
    $b = trim(fgets(STDIN));
  }

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

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

    if ($d1 == $d4 && $d2 == $d3) {
      echo $x;
    }

  }
?>
```

17. Solution

```
<?php
  for ($i = 0; $i <= 30; $i++) {
    echo pow(2, $i), "\n";
  }
?>
```

18. Solution

```
<?php
  $offset = 10;
  $i = 1;
  while ($i <= 401) {
    echo $i ,"\n";
    $i += $offset;
    $offset += 2;
  }
?>
```

19. Solution

```
<?php
```

```
  for ($i = 1; $i <= 100; $i++) {
    echo -$i, "\n", $i, "\n";
  }
?>
```

20. Solution

First Approach

```
<?php
  $value = 0;
  for ($i = 1; $i <= 8; $i++) {
    $offset = pow(10, $i - 1);
    $value += $offset;
    echo $value, "\n";
  }
?>
```

Second Approach

```
<?php
  $value = "1";
  for ($i = 1; $i <= 8; $i++) {
    echo $value, "\n";
    $value .= "1";
  }
?>
```

21. Solution

```
<?php
  $a = trim(fgets(STDIN));

  $fib_prev_prev = 0;
  $fib_prev = 1;
  $fib = 1;
  for($i = 1;$i <= $a; $i++) {
    echo $fib . "\n";
    $fib = $fib_prev + $fib_prev_prev;
    $fib_prev_prev = $fib_prev;
    $fib_prev = $fib;
  }
?>
```

22. Solution

```
<?php
  $a = trim(fgets(STDIN));

  $fib_prev_prev = 0;
  $fib_prev = 1;
  $fib = 1;
  while ($fib < $a) {
    echo $fib . "\n";
    $fib = $fib_prev + $fib_prev_prev;
    $fib_prev_prev = $fib_prev;
    $fib_prev = $fib;
  }
?>
```

23. Solution

```
<?php
  echo "Enter a positive integer: ";
  $n = trim(fgets(STDIN));
  while (is_numeric($n) != true || $n < 1) {
    echo "Wrong number. Please enter a positive integer: ";
    $n = trim(fgets(STDIN));
  }

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

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

  $y = $nominator / $denominator;
  echo $y;
?>
```

24. Solution

```
<?php
  echo "Enter a positive integer: ";
  $n = trim(fgets(STDIN));
  while (is_numeric($n) != true || $n < 1) {
    echo "Wrong number. Please enter a positive integer: ";
    $n = trim(fgets(STDIN));
  }

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

  $y = $nominator / $n;
  echo $y;
?>
```

25. Solution

```
<?php
  echo "Enter a positive integer: ";
  $n = trim(fgets(STDIN));
  while (is_numeric($n) != true || $n < 1) {
    echo "Wrong number. Please enter a positive integer: ";
    $n = trim(fgets(STDIN));
  }

  $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 / $i;
    $sign = -$sign;
  }

  echo $y;
?>
```

26. Solution

```
<?php
  echo "Enter a positive integer: ";
  $n = trim(fgets(STDIN));
  while (is_numeric($n) != true || $n < 1) {
    echo "Wrong number. Please enter a positive integer: ";
    $n = trim(fgets(STDIN));
  }

  $y = 0;
  for ($i = 1; $i <= $n; $i++) {
    $y += 1 / pow($i, $n - $i + 1);
  }

  echo $y;
?>
```

27. Solution

```
<?php
  echo "Enter a non-negative integer: ";
  $n = trim(fgets(STDIN));

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

  echo $factorial;
?>
```

***Notice**: Please note that this PHP script operates properly for all non-negative integers, including zero.*

28. Solution

First Approach

```
<?php
  define("ACCURACY", 0.00001);

  $x = trim(fgets(STDIN));

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

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

    $exponential += pow($x, $i) / $factorial;

    $i++;
  } while (abs($exponential - $exponential_previous) > ACCURACY);

  echo "e(", $x ,") ~= ", $exponential;
?>
```

Second Approach

```
<?php
  define("ACCURACY", 0.00001);

  $x = trim(fgets(STDIN));
  $exponential = 1;
  $i = 1;
  $factorial = 1;
  do {
    $exponential_previous = $exponential;

    $factorial *= $i;

    $exponential += pow($x, $i) / $factorial;

    $i++;
  } while (abs($exponential - $exponential_previous) > ACCURACY);

  echo "e(", $x ,") ~= ", $exponential;
?>
```

29. Solution

First Approach

```
<?php
  define("ACCURACY", 0.00001);

  $x = trim(fgets(STDIN));

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

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

    $sinus += $sign * pow($x, $i) / $factorial;

    $sign = -$sign;
    $i += 2;
  } while (abs($sinus - $sinus_previous) > ACCURACY);

  echo "sin(", $x ,") ~= ", $sinus;
?>
```

Second Approach

```
<?php
```

```
  define("ACCURACY", 0.00001);

  $x = trim(fgets(STDIN));

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

    $factorial *= $i * ($i - 1);

    $sinus += $sign * pow($x, $i) / $factorial;

    $sign = -$sign;
    $i += 2;
  } while (abs($sinus - $sinus_previous) > ACCURACY);

  echo "sin(", $x ,") ~= ", $sinus;
?>
```

30. Solution

First Approach

```
<?php
  define("ACCURACY", 0.00001);

  $x = trim(fgets(STDIN));

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

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

    $cosinus += $sign * pow($x, $i) / $factorial;

    $sign = -$sign;
    $i += 2;
  } while (abs($cosinus - $cosinus_previous) > ACCURACY);
  echo "cos(", $x ,") ~= ", $cosinus;
?>
```

Second Approach

```
<?php
  define("ACCURACY", 0.00001);

  $x = trim(fgets(STDIN));

  $sign = -1;
  $cosinus = 1;
  $i = 2;
  $factorial = 1;
  do {
    $cosinus_previous = $cosinus;

    $factorial *= $i * ($i - 1);

    $cosinus += $sign * pow($x, $i) / $factorial;

    $sign = -$sign;
    $i += 2;
  } while (abs($cosinus - $cosinus_previous) > ACCURACY);

  echo "cos(", $x ,") ~= ", $cosinus;
?>
```

31. Solution

```
<?php
  $max = -460;
  $sum = 0;
  for ($i = 1; $i <= 31; $i++) {
    do {
      echo "Enter temperature for day ", $i, ": ";
      $t = trim(fgets(STDIN));

      $failure = false;
      if (is_numeric($t) != true) {
        echo "Please enter numeric values! \n";
        $failure = true;
      }
      elseif ($t < -459.67) {
        echo "Please enter a value greater than 459.67\n";
        $failure = true;
      }
    } while ($failure == true);
    $sum += $t;
    if ($t > $max) {
      $max = $t;
    }
  }

  echo $sum / 31, $max;
?>
```

32. Solution

```
<?php
  $level = trim(fgets(STDIN));
  if ($level != 9999) {
    $hour = trim(fgets(STDIN));
    $minutes = trim(fgets(STDIN));

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

    $min = $level;
    $min_hour = $hour;
    $min_minutes = $minutes;

    $level = trim(fgets(STDIN));
    while ($level != 9999) {
      $hour = trim(fgets(STDIN));
      $minutes = trim(fgets(STDIN));

      if ($level > $max) {
        $max = $level;
        $max_hour = $hour;
        $max_minutes = $minutes;
      }

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

      $level = trim(fgets(STDIN));
    }

    echo $max, $max_hour, $max_minutes;
    echo $min, $min_hour, $min_minutes;
  }
?>
```

33. Solution

```
<?php
  $alphabet = "abcdefghijklmnopqrstuvwxyz";
  do {
    echo "Enter an integer between 1 and 26: ";
    $a = trim(fgets(STDIN));

    $failure = false;
    if (is_numeric($a) != true) {
      echo "Please enter numeric values! \n";
      $failure = true;
    }
    elseif ($a < 1) {
      echo "Please enter positive integers! \n";
      $failure = true;
    }
    elseif ($a > 26) {
      echo "Please enter a value less than or equal to 26! \n";
      $failure = true;
    }
  } while ($failure == true);

  do {
    echo "Enter an integer between 1 and 26: ";
    $b = trim(fgets(STDIN));

    $failure = false;
    if (is_numeric($b) != true) {
      echo "Please enter numeric values! \n";
      $failure = true;
    }
    elseif ($b < 1) {
      echo "Please enter positive integers! \n";
      $failure = true;
    }
    elseif ($b > 26) {
      echo "Please enter a value less than or equal to 26! \n";
      $failure = true;
    }
  } while ($failure == true);

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

  for ($i = $a; $i <= $b; $i++) {
    echo $alphabet[$i - 1];
  }
?>
```

34. Solution

```
<?php
  $secret_number = rand(1, 100);

  $attempts = 1;
  echo "Enter a guess: ";
  $guess = trim(fgets(STDIN));
  while ($guess != $secret_number) {
    if ($guess > $secret_number) {
      echo "Your guess is bigger than my secret number. Try again.\n";
    }
    else {
      echo "Your guess is smaller than my secret number. Try again.\n";
    }
    $attempts++;
    echo "Enter a guess: ";
    $guess = trim(fgets(STDIN));
  }
  echo "You found it!\n";
  echo "Attempts: ", $attempts;
?>
```

35. Solution

```
<?php
  for ($i = 1; $i <= 2; $i++) {
    $secret_number = rand(1, 100);

    $attempts = 1;
    echo "Enter a guess: ";
    $guess = trim(fgets(STDIN));
    while ($guess != $secret_number) {
      if ($guess > $secret_number) {
        echo "Your guess is bigger than my secret number. Try again.\n";
      }
      else {
        echo "Your guess is smaller than my secret number. Try again.\n";
      }
      $attempts++;
      echo "Enter a guess: ";
      $guess = trim(fgets(STDIN));
    }
    echo "You found it!\n";
    echo "Attempts: ", $attempts, "\n";

    if ($i == 1) {
      $first_player_attempts = $attempts;
    }
  }

  if ($first_player_attempts < $attempts) {
    echo "First player wins!";
  }
  elseif ($first_player_attempts > $attempts) {
    echo "Second player wins!";
  }
  else {
    echo "It's a draw";
  }
?>
```

36. Solution

```
<?php
  do {
    echo "1. 4/3 TV Screen\n";
    echo "2. 16/9 TV Screen\n";
    echo "3. Exit\n";
    echo "Enter a choice: ";
    $choice = trim(fgets(STDIN));

    if ($choice == 1) {
      echo "Enter diagonal: ";
      $diagonal = trim(fgets(STDIN));
      echo "Width: ", $diagonal * 0.8, "\n";
      echo "Height: ", $diagonal * 0.6, "\n";
    }
    elseif ($choice == 2) {
      echo "Enter diagonal: ";
      $diagonal = trim(fgets(STDIN));
      echo "Width: ", $diagonal * 0.87, "\n";
      echo "Height: ", $diagonal * 0.49, "\n";
    }
  } while ($choice != 3);
?>
```

37. Solution

```
<?php
  echo "Enter total number of students: ";
  $n = trim(fgets(STDIN));
  while (is_numeric($n) != true || $n < 1) {
    echo "Wrong number. Please enter total number of students: ";
    $n = trim(fgets(STDIN));
  }

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

  $max = -1;
  $min = 101;

  for ($i = 1; $i <= $n; $i++) {
    echo "Enter grade for student No ", $i, ": ";
    $grade = trim(fgets(STDIN));
    while (is_numeric($grade) != true || $grade < 0 || $grade > 100) {
      echo "Wrong grade. Please enter grade for student No ", $i, ": ";
      $grade = trim(fgets(STDIN));
    }

    echo "Enter sex for student No ", $i, ": ";
    $sex = strtoupper(trim(fgets(STDIN)));
    while ($sex != "M" && $sex != "F") {
      echo "Wrong sex. Please enter sex for student No ", $i, ": ";
      $sex = strtoupper(trim(fgets(STDIN)));
    }

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

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

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

    $sum += $grade;
  }

  if ($count_a > 0) {
    echo "The average value of those who got an 'A' is: ", $sum_a / $count_a, "\n";
  }
  if ($count_b > 0) {
    echo "The average value of those who got a 'B' is: ", $sum_b / $count_b, "\n";
  }
  if ($count_a_boys > 0) {
    echo "The average value of boys who got an 'A' is: ", $sum_a_boys / $count_a_boys, "\n";
  }
  echo "The total number of girls that got less than 'B' is: ", $count_cdef_girls, "\n";
  echo "The highest grade is: ", $max, "\n";
  echo "The lowest grade is: ", $min, "\n";
  echo "The average grade of the whole class is: ", $sum / $n, "\n";
?>
```

38. Solution

```
<?php
  do {
    echo "Enter amount: ";
    $amount = trim(fgets(STDIN));
    while (is_numeric($amount) != true || $amount < 0) {
      echo "Wrong amount. Please enter amount: ";
      $amount = trim(fgets(STDIN));
    }
    if ($amount < 20) {
      $discount = 0;
    }
    elseif ($amount < 50) {
      $discount = 3;
    }
    elseif ($amount < 100) {
      $discount = 5;
    }
    else {
      $discount = 10;
    }

    echo "Discount: ", $discount, "%\n";
    echo "Amount to pay (discount included): ", $amount - $amount * $discount / 100, "\n";

    echo "Would you like to repeat? ";
    $answer = trim(fgets(STDIN));
  } while (strtoupper($answer) == "YES");
?>
```

39. Solution

```
<?php
  define("TAX_RATE", 0.25);

  echo "Enter number of Kilowatt-hours consumed: ";
  $kwh = trim(fgets(STDIN));
  while (is_numeric($kwh) != true || $kwh < 0 && $kwh != -1) {
    echo "Wrong value. Please enter number of Kilowatt-hours consumed: ";
    $kwh = trim(fgets(STDIN));
  }

  while ($kwh != -1) {
    if ($kwh <= 400) {
      $t = $kwh * 0.11;
    }
    elseif ($kwh <= 1500) {
      $t = 400 * 0.11 + ($kwh - 400) * 0.22;
    }
    elseif ($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;
    echo "Total amount to pay (taxes included): ", $t, "\n";

    echo "Enter number of Kilowatt-hours consumed: ";
    $kwh = trim(fgets(STDIN));
    while (is_numeric($kwh) != true || $kwh < 0 && $kwh != -1) {
      echo "Wrong value. Please enter number of Kilowatt-hours consumed: ";
      $kwh = trim(fgets(STDIN));
    }
  }
?>
```

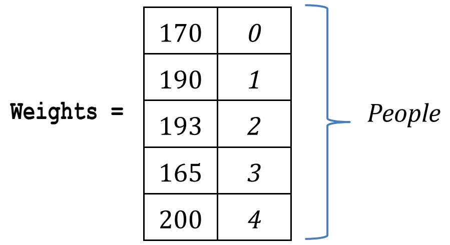
Chapter 32

32.3 Answers of Review Questions: True/False

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

32.4 Answers of Review Exercises

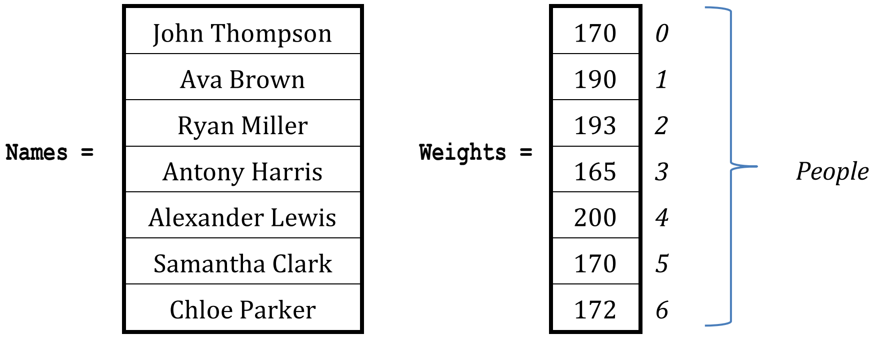
1. Solution



Weights =

170	*0*
190	*1*
193	*2*
165	*3*
200	*4*

2. Solution



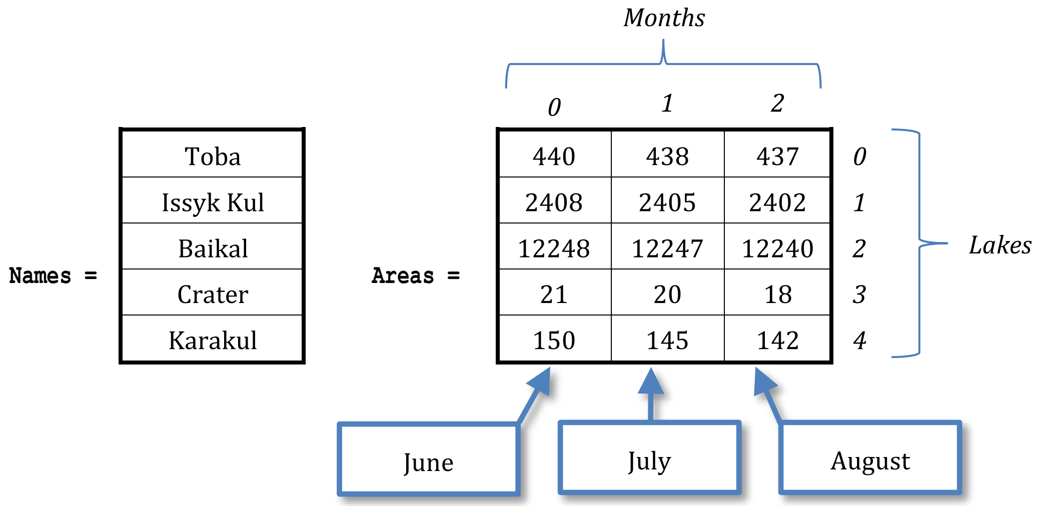
Names =

John Thompson
Ava Brown
Ryan Miller
Antony Harris
Alexander Lewis
Samantha Clark
Chloe Parker

Weights =

170	*0*
190	*1*
193	*2*
165	*3*
200	*4*
170	*5*
172	*6*

3. Solution



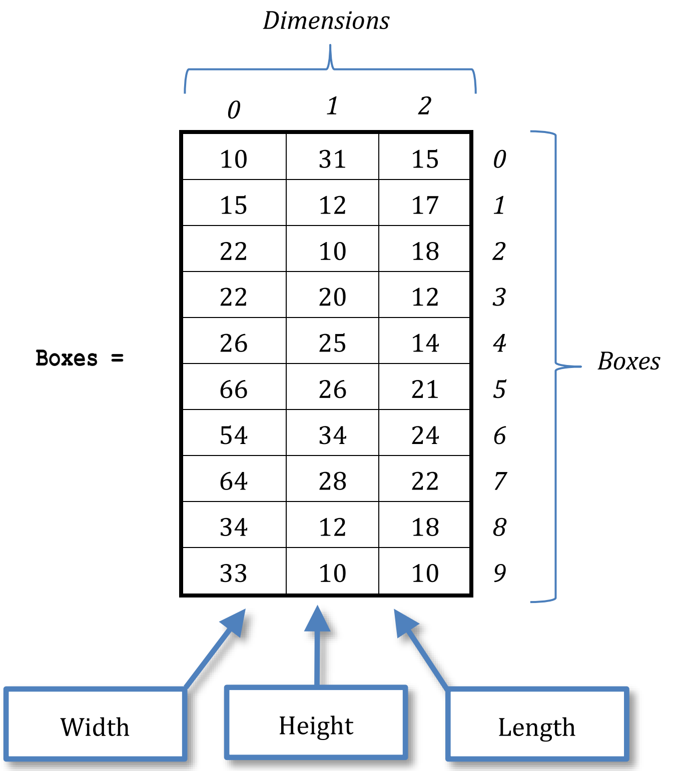
Names =

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*

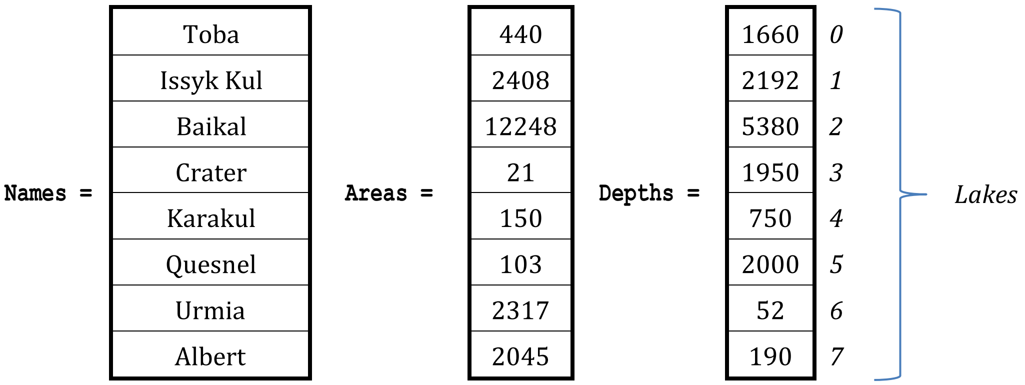
4. Solution



Boxes =

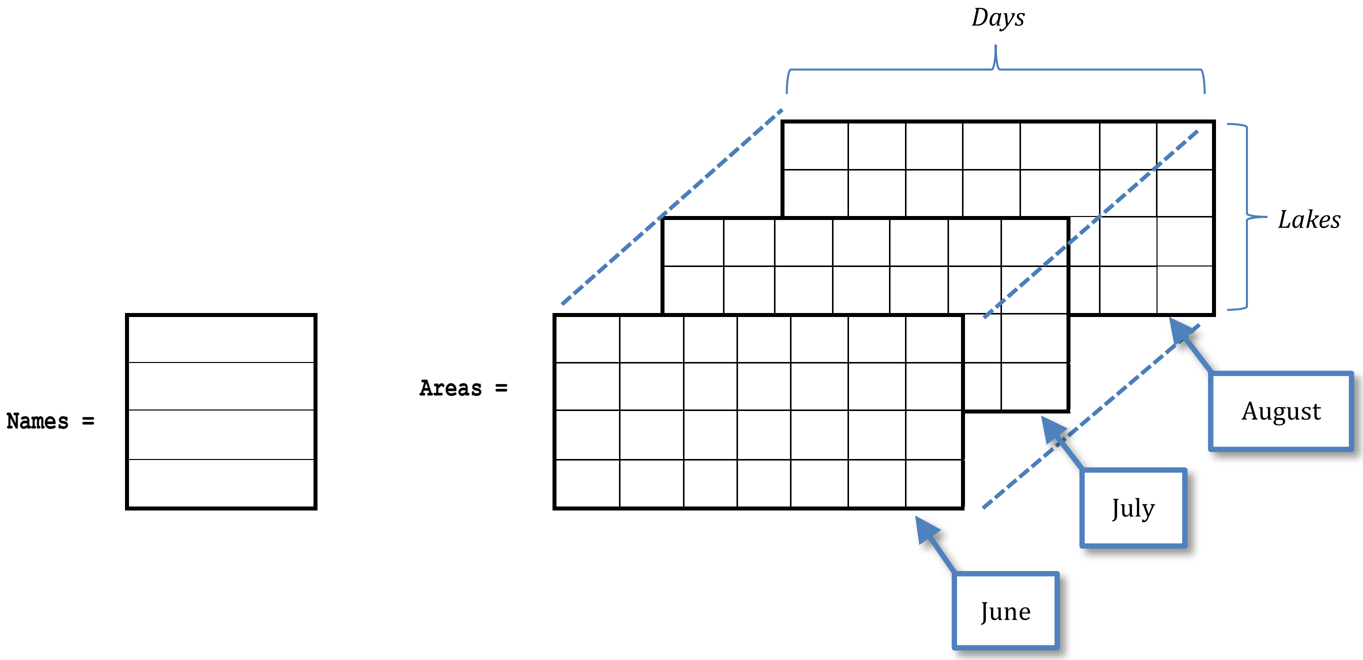
0	1	2	
10	31	15	*0*
15	12	17	*1*
22	10	18	*2*
22	20	12	*3*
26	25	14	*4*
66	26	21	*5*
54	34	24	*6*
64	28	22	*7*
34	12	18	*8*
33	10	10	*9*

5. Solution



Names =	Areas =	Depths =	
Toba	440	1660	*0*
Issyk Kul	2408	2192	*1*
Baikal	12248	5380	*2*
Crater	21	1950	*3*
Karakul	150	750	*4*
Quesnel	103	2000	*5*
Urmia	2317	52	*6*
Albert	2045	190	*7*

6. Solution



Chapter 33

33.5 Answers of Review Questions: True/False

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

33.6 Answers of Review Questions: Multiple Choice

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

33.7 Answers of Review Exercises

1. Solution

Step	Statement	$x	$a[0]	$a[1]	$a[2]
1	`$a = array()`	?	?	?	?
2	`$a[2] = 1`	?	?	?	**1**
3	`$x = 0`	**0**	?	?	1
4	`$a[$x + $a[2]] = 4`	0	?	**4**	1
5	`$a[$x] = $a[$x + 1] * 4`	0	**16**	4	1

2. Solution

Step	Statement	$x	$a[0]	$a[1]	$a[2]	$a[3]	$a[4]
1	`$a = array()`	?	?	?	?	?	?
2	`$a[1] = 5`	?	?	5	?	?	?
3	`$x = 0`	0	?	5	?	?	?
4	`$a[$x] = 4`	0	4	5	?	?	?
5	`$a[$a[0]] = $a[$x + 1] % 3`	0	4	5	?	?	2

6	`$a[$a[0] / 2] = 10`	0	4	5	10	?	2
7	`$x += 2`	2	4	5	10	?	2
8	`$a[$x + 1] = $a[$x] + 9`	2	4	5	10	19	2

3. Solution

For input value of 3

Step	Statement	\$x	\$a[0]	\$a[1]	\$a[2]	\$a[3]
1	`$a = array()`	?	?	?	?	?
2	`$a[1] = trim(fgets(STDIN))`	?	?	**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	`$a = array()`	?	?	?	?	?
2	`$a[1] = trim(fgets(STDIN))`	?	?	**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	`$a = array()`	?	?	?	?	?
2	`$a[1] = trim(fgets(STDIN))`	?	?	**1**	?	?
3	`$x = 0`	**0**	?	1	?	?
4	`$a[$x] = 3`	0	**3**	1	?	?
5	`$a[$a[0]] = $a[$x + 1] % 2`	0	3	1	?	**3**
6	`$a[$a[0] % 2] = 10`	0	3	**10**	?	3
7	`$x++`	**1**	3	10	?	3
8	`$a[$x + 1] = $a[$x] + 9`	1	3	10	**19**	3

4. Solution

For input value of 100

Step	Statement	$x	$a[0]	$a[1]	$a[2]	$a[3]
1	`$a = array()`	?	?	?	?	?
2	`$a[1] = trim(fgets(STDIN))`	?	?	**100**	?	?
3	`$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]	$a[3]
1	`$a = array()`	?	?	?	?	?
2	`$a[1] = trim(fgets(STDIN))`	?	?	**108**	?	?
3	`$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]	$a[3]
1	`$a = array()`	?	?	?	?	?
2	`$a[1] = trim(fgets(STDIN))`	?	?	**1**	?	?
3	`$x = 0`	**0**	?	1	?	?
4	`$a[$x] = 3`	0	**3**	1	?	?
5	`$a[$a[0]] = $a[$x + 1] % 10`	0	3	1	?	**1**
6	`if ($a[3] > 5)`	False				
7	`$a[2] = 3`	0	3	1	**3**	1

5. Solution

Step	Statement	$x	$y	$a[0]	$a[1]	$a[2]
1	`$a = array()`	?	?	?	?	?
2	`$x = 4`	**4**	?	?	?	?
3	`$y = $x - 1`	4	**3**	?	?	?
4, 5	`if ($x > $y)`	4	3	**1**	?	?

	$a[0] = 1; else $a[0] = $y;					
6	$a[1] = $x + 3	4	3	1	**7**	?
7	$y = $y - 1	4	**2**	1	7	?
8	$a[$y] = ($x + 5) % 2	4	2	1	7	**1**

6. Solution

Step	Statement	$i	$a[0]	$a[1]	$a[2]	$a[3]	$a[4]	$a[5]
1	a = array(17, 12, 45, 12, 12, 49)	?	**17**	**12**	**45**	**12**	**12**	**49**
2	$i = 0;	**0**	17	12	45	12	12	49
3	$i <= 5	True						
4	if ($a[$i] == 12)	False						
5	$a[$i]++	0	**18**	12	45	12	12	49
6	$i++	**1**	18	12	45	12	12	49
7	$i <= 5	True						
8	if ($a[$i] == 12)	True						
9	$a[$i]--	1	18	**11**	45	12	12	49
10	$i++	**2**	18	11	45	12	12	49
11	$i <= 5	True						
12	if ($a[$i] == 12)	False						
13	$a[$i]++	2	18	11	**46**	12	12	49
14	$i++	**3**	18	11	46	12	12	49
15	$i <= 5	True						
16	if ($a[$i] == 12)	True						
17	$a[$i]--	3	18	11	46	**11**	12	49
18	$i++	**4**	18	11	46	11	12	49
19	$i <= 5	True						
20	if ($a[$i] == 12)	True						
21	$a[$i]--	4	18	11	46	11	**11**	49
22	$i++	**5**	18	11	46	11	11	49
23	$i <= 5	True						
24	if ($a[$i] == 12)	False						
25	$a[$i]++	5	18	11	46	11	11	**50**
26	$i++	**6**	18	11	46	11	11	50
27	$i <= 5	False						

7. Solution

Step	Statement	$i	$a[0]	$a[1]	$a[2]	$a[3]	$a[4]	$a[5]
1	`$a = array(10, 15, 12, 23, 22, 19)`	?	**10**	**15**	**12**	**23**	**22**	**19**
2	`$i = 1`	**1**	10	15	12	23	22	19
3	`$i <= 4`	True						
4	`$a[$i] = $a[$i + 1] + $a[$i - 1]`	1	10	**22**	12	23	22	19
5	`$i++`	**2**	10	22	12	23	22	19
6	`$i <= 4`	True						
7	`$a[$i] = $a[$i + 1] + $a[$i - 1]`	2	10	22	**45**	23	22	19
8	`$i++`	**3**	10	22	45	23	22	19
9	`$i <= 4`	True						
10	`$a[$i] = $a[$i + 1] + $a[$i - 1]`	3	10	22	45	**67**	22	19
11	`$i++`	**4**	10	22	45	67	22	19
12	`$i <= 4`	True						
13	`$a[$i] = $a[$i + 1] + $a[$i - 1]`	4	10	22	45	67	**86**	19
14	`$i++`	**5**	10	22	45	67	86	19
15	`$i <= 4`	False						

8. Solution

```
<?php
  define("ELEMENTS", 100);

  $a = array();
  for ($i = 0; $i <= ELEMENTS - 1; $i++) {
    $a[$i] = trim(fgets(STDIN));
  }

  for ($i = 0; $i <= ELEMENTS - 1; $i++) {
    echo pow($a[$i], 3);
  }
?>
```

9. Solution

```
<?php
  define("ELEMENTS", 80);

  $a = array();
  for ($i = 0; $i <= ELEMENTS - 1; $i++) {
    $a[$i] = trim(fgets(STDIN));
  }
  for ($i = ELEMENTS - 1; $i >= 0; $i--) {
    echo pow($a[$i], 2);
  }
?>
```

10. Solution

```
<?php
  define("ELEMENTS", 90);

  $a = array();
  for ($i = 0; $i <= ELEMENTS - 1; $i++) {
    $a[$i] = trim(fgets(STDIN));
  }

  for ($i = ELEMENTS - 1; $i >= 0; $i--) {
    if ($a[$i] % 5 == 0) {
      echo $a[$i];
    }
  }
?>
```

11. Solution

```
<?php
  define("ELEMENTS", 50);

  $a = array();
  for ($i = 0; $i <= ELEMENTS - 1; $i++) {
    $a[$i] = trim(fgets(STDIN));
  }

  for ($i = 0; $i <= ELEMENTS - 1; $i++) {
    if ($a[$i] % 2 == 0 || $a[$i] > 10) {
      echo $a[$i];
    }
  }
?>
```

12. Solution

```
<?php
  define("ELEMENTS", 30);

  $a = array();
  for ($i = 0; $i <= ELEMENTS - 1; $i++) {
    $a[$i] = trim(fgets(STDIN));
  }

  $sum = 0;
  for ($i = 0; $i <= ELEMENTS - 1; $i++) {
    if ($a[$i] > 0) {
      $sum += $a[$i];
    }
  }
  echo $sum;
?>
```

13. Solution

```
<?php
  define("ELEMENTS", 50);

  $a = array();
  for ($i = 0; $i <= ELEMENTS - 1; $i++) {
    $a[$i] = trim(fgets(STDIN));
  }

  $sum = 0;
  for ($i = 0; $i <= ELEMENTS - 1; $i++) {
    if ($a[$i] >= 10 && $a[$i] <= 99) {
      $sum += $a[$i];
    }
  }
  echo $sum;
?>
```

14. Solution

```
<?php
  define("ELEMENTS", 40);

  $a = array();
  for ($i = 0; $i <= ELEMENTS - 1; $i++) {
    $a[$i] = trim(fgets(STDIN));
  }

  $sum_pos = 0;
  $sum_neg = 0;
  for ($i = 0; $i <= ELEMENTS - 1; $i++) {
    if ($a[$i] > 0 ) {
      $sum_pos += $a[$i];
    }
    elseif ($a[$i] < 0) {
      $sum_neg += $a[$i];
    }
  }
  echo $sum_pos, $sum_neg;
?>
```

15. Solution

```php
<?php
  define("ELEMENTS", 20);

  $a = array();
  for ($i = 0; $i <= ELEMENTS - 1; $i++) {
    $a[$i] = trim(fgets(STDIN));
  }

  $sum = 0;
  for ($i = 0; $i <= ELEMENTS - 1; $i++) {
    $sum += $a[$i];
  }
  echo $sum / ELEMENTS;
?>
```

16. Solution

```php
<?php
  define("WORDS", 50);

  $a = array();
  for ($i = 0; $i <= WORDS - 1; $i++) {
    $a[$i] = trim(fgets(STDIN));
  }

  for ($i = 0; $i <= WORDS - 1; $i++) {
    if (strlen($a[$i]) >= 10 ) {
      echo $a[$i];
    }
  }
?>
```

17. Solution

```php
<?php
  define("WORDS", 40);

  $a = array();
  for ($i = 0; $i <= WORDS - 1; $i++) {
    $a[$i] = trim(fgets(STDIN));
  }

  for ($i = 0; $i <= WORDS - 1; $i++) {
    $count = 0;
    for ($j = 0; $j <= strlen($a[$i]) - 1; $j++) {
      if (substr($a[$i], $j, 1) == "w") {
        $count++;
      }
    }
    if ($count >= 2) {
      echo $a[$i], "\n";
    }
  }
?>
```

Chapter 34

34.7 Answers of Review Questions: True/False

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

34.8 Answers of Review Questions: Multiple Choice

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

34.9 Answers of Review Exercises

1. Solution

Step	Statement	$x	$a
1	`$a = array(array());`	?	? ? ? ? ? ?
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	$a
1	`$a = array(array());`	?	?	? | ? | ? ? | ? | ?
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 + $j`	1	0	5 **10**	6 ?	7 ?
20	`$j++`	1	**1**	5 10	6 ?	7 ?
21	`$j <= 2`	True				
22	`$a[$i][$j] = ($i + 1) * 5 + $j`	1	1	5 10	6 **11**	7 ?
23	`$j++`	1	**2**	5 10	6 11	? ?
24	`$j <= 2`	True				
25	`$a[$i][$j] = ($i + 1) * 5 + $j`	1	2	5 10	6 11	? **12**
26	`$j++`	1	**3**	5 10	6 11	7 12
27	`$j <= 2`	False				

3. Solution

Step	Statement	$i	$j	$a
1	`$a = array(array());`	?	?	? ? ? ? ? ? ? ? ?
2	`$j = 0`	?	**0**	? ? ? ? ? ? ? ? ?
3	`$j <= 2`	True		
4	`$i = 0`	0	**0**	? ? ? ? ? ? ? ? ?
5	`$i <= 2`	True		
6	`$a[$i][$j] = ($i + 1) * 2 + $j * 4`	0	0	**2** ? ? ? ? ? ? ? ?
7	`$i++`	**1**	0	2 ? ? ? ? ? ? ? ?
8	`$i <= 2`	True		
9	`$a[$i][$j] = ($i + 1) * 2 + $j * 4`	1	0	2 ? ? **4** ? ? 6 ? ?
10	`$i++`	**2**	0	2 ? ? 4 ? ? 6 ? ?
11	`$i <= 2`	True		
12	`$a[$i][$j] = ($i + 1) * 2 + $j * 4`	2	0	2 ? ? 4 ? ? **6** ? ?
13	`$i++`	**3**	0	2 ? ? 4 ? ? 6 ? ?
14	`$i <= 2`	False		

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

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

4. Solution

For input value of 5

0	5	10
0	6	12

For input value of 9

0	9	18
0	10	20

For input value of 3

0	3	6
0	4	8

5. Solution

For input value of 13

0	3	3
0	17	18

For input value of 10

0	10	3
0	11	15

For input value of 8

3	3	3
11	12	13

6. Solution

19	5	31
28	6	20

7. Solution

26	29
37	34
59	49

8. Solution

i. -1 15 22 25 12 16 7 9 1
ii. 7 9 1 25 12 16 -1 15 22
iii. 22 15 -1 16 12 25 1 9 7
iv. 1 9 7 16 12 25 22 15 -1
v. -1 25 7 15 12 9 22 16 1
vi. 7 25 -1 9 12 15 1 16 22
vii. 22 16 1 15 12 9 -1 25 7
viii. 1 16 22 9 12 15 7 25 -1

9. Solution

```
<?php
  define("ROWS", 10);
  define("COLUMNS", 15);

  $a = array(array());
  for ($i = 0; $i <= ROWS - 1; $i++) {
    for ($j = 0; $j <= COLUMNS - 1; $j++) {
      $a[$i][$j] = trim(fgets(STDIN));
    }
  }

  for ($i = 0; $i <= ROWS - 1; $i++) {
    for ($j = 0; $j <= COLUMNS - 1; $j++) {
      if ($a[$i][$j] % 2 != 0) {
        echo $i, $j;
      }
    }
  }
?>
```

10. Solution

```
<?php
  define("ROWS", 10);
  define("COLUMNS", 6);

  $a = array(array());
  for ($i = 0; $i <= ROWS - 1; $i++) {
    for ($j = 0; $j <= COLUMNS - 1; $j++) {
      $a[$i][$j] = trim(fgets(STDIN));
    }
  }

  for ($i = 0; $i <= ROWS - 1; $i++) {
    for ($j = 0; $j <= COLUMNS - 1; $j += 2) {
      echo $a[$i][$j];
    }
  }
?>
```

11. Solution

```
<?php
  define("ROWS", 12);
  define("COLUMNS", 8);
  $a = array(array());
  for ($i = 0; $i <= ROWS - 1; $i++) {
    for ($j = 0; $j <= COLUMNS - 1; $j++) {
      $a[$i][$j] = trim(fgets(STDIN));
    }
  }

  $sum = 0;
  for ($i = 1; $i <= ROWS - 1; $i += 2) {
    for ($j = 0; $j <= COLUMNS - 1; $j += 2) {
      $sum += $a[$i][$j];
    }
  }
  echo $sum;
?>
```

12. Solution

```
<?php
  define("N", 8 );

  $a = array(array());
  for ($i = 0; $i <= N - 1; $i++) {
    for ($j = 0; $j <= N - 1; $j++) {
      $a[$i][$j] = trim(fgets(STDIN));
    }
  }

  $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];
  }
  echo $sum_diagonal / N, $sum_antidiagonal / N;
?>
```

13. Solution

```
<?php
  define("N", 5);

  $a = array(array());
  for ($i = 0; $i <= N - 1; $i++) {
    for ($j = 0; $j <= N - 1; $j++) {
      if ($i == N - $j - 1) {
        $a[$i][$j] = 5;
      }
      elseif ($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++) {
      echo $a[$i][$j], "\t";
    }
    echo "\n";
  }
?>
```

14. Solution

```
<?php
  define("N", 5);

  $a = array(array());
  for ($i = 0; $i <= N - 1; $i++) {
    for ($j = 0; $j <= N - 1; $j++) {
      if ($i == N - $j - 1) {
        $a[$i][$j] = 5;
      }
      elseif ($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++) {
      echo $a[$i][$j], "\t";
    }
    echo "\n";
  }
?>
```

15. Solution

```
<?php
  define("ROWS", 5);
  define("COLUMNS", 4);

  $a = array(array());
  for ($i = 0; $i <= ROWS - 1; $i++) {
    for ($j = 0; $j <= COLUMNS - 1; $j++) {
      $a[$i][$j] = trim(fgets(STDIN));
    }
  }

  for ($i = 0; $i <= ROWS - 1; $i++) {
    for ($j = 0; $j <= COLUMNS - 1; $j++) {
      if ($a[$i][$j] == intval($a[$i][$j])) {
        echo $i, $j;
      }
    }
  }
?>
```

16. Solution

```
<?php
  define("ROWS", 10);
  define("COLUMNS", 4);

  $a = array(array());
  for ($i = 0; $i <= ROWS - 1; $i++) {
    for ($j = 0; $j <= COLUMNS - 1; $j++) {
      $a[$i][$j] = trim(fgets(STDIN));
    }
  }

  $count = 0;
  for ($i = 0; $i <= ROWS - 1; $i++) {
    for ($j = 0; $j <= COLUMNS - 1; $j++) {
      if ($a[$i][$j] < 0) {
        $count++;
      }
    }
  }
  echo $count;
```

```
?>
```

17. Solution

```
<?php
  define("ROWS", 3);
  define("COLUMNS", 4);

  $a = array(array());
  for ($i = 0; $i <= ROWS - 1; $i++) {
    for ($j = 0; $j <= COLUMNS - 1; $j++) {
      $a[$i][$j] = trim(fgets(STDIN));
    }
  }

  for ($i = 0; $i <= ROWS - 1; $i++) {
    for ($j = 0; $j <= COLUMNS - 1; $j++) {
      echo $a[$i][$j], " ";
    }
  }
?>
```

18. Solution

```
<?php
  define("ROWS", 20);
  define("COLUMNS", 14);

  $a = array(array());
  for ($i = 0; $i <= ROWS - 1; $i++) {
    for ($j = 0; $j <= COLUMNS - 1; $j++) {
      $a[$i][$j] = trim(fgets(STDIN));
    }
  }

  for ($i = 0; $i <= ROWS - 1; $i++) {
    for ($j = 0; $j <= COLUMNS - 1; $j++) {
      if (strlen($a[$i][$j]) < 5) {
        echo $a[$i][$j], " ";
      }
    }
  }
?>
```

19. Solution

First Approach

```
<?php
  define("ROWS", 20);
  define("COLUMNS", 14);

  $a = array(array());
  for ($i = 0; $i <= ROWS - 1; $i++) {
    for ($j = 0; $j <= COLUMNS - 1; $j++) {
      $a[$i][$j] = trim(fgets(STDIN));
    }
  }

  $length_limits = array(5, 10, 20);

  for ($k = 0; $k <= 2; $k++) {
    for ($i = 0; $i <= ROWS - 1; $i++) {
      for ($j = 0; $j <= COLUMNS - 1; $j++) {
        if (strlen($a[$i][$j]) < $length_limits[$k]) {
          echo $a[$i][$j], " ";
        }
      }
    }
  }
?>
```

Second Approach

```
<?php
  define("ROWS", 20);
  define("COLUMNS", 14);

  $a = array(array());
  for ($i = 0; $i <= ROWS - 1; $i++) {
    for ($j = 0; $j <= COLUMNS - 1; $j++) {
      $a[$i][$j] = trim(fgets(STDIN));
    }
  }

  for ($k = 0; $k <= 2; $k++) {
    for ($i = 0; $i <= ROWS - 1; $i++) {
      for ($j = 0; $j <= COLUMNS - 1; $j++) {
        if (strlen($a[$i][$j]) < 5 * pow(2, $k)) {
          echo $a[$i][$j], " ";
        }
      }
    }
    }
?>
```

Chapter 35

35.7 Answers of Review Questions: True/False

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

35.8 Answers of Review Questions: Multiple Choice

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

35.9 Answers of Review Exercises

1. Solution

```
<?php
  define("STUDENTS", 15);
  define("TESTS", 5);

  $grades = array(array());
  for ($i = 0; $i <= STUDENTS - 1; $i++) {
    for ($j = 0; $j <= TESTS - 1; $j++) {
      $grades[$i][$j] = trim(fgets(STDIN));
    }
  }

  $average = array();
  for ($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++) {
    echo "Student No ", $i + 1, ": ";

    if ($average[$i] < 60) {
      echo "E/F";
    }
    elseif ($average[$i] < 70) {
      echo "D";
    }
    elseif ($average[$i] < 80) {
      echo "C";
    }
    elseif ($average[$i] < 90) {
      echo "B";
    }
    else {
      echo "A";
    }

    echo "\n";
  }
?>
```

2. Solution

```
<?php
  define("OBJECTS", 5);
  define("FALLS", 10);

  $g = array(array());
  for ($i = 0; $i <= OBJECTS - 1; $i++) {
    for ($j = 0; $j <= FALLS - 1; $j++) {
      $g[$i][$j] = trim(fgets(STDIN));
    }
  }

  for ($i = 0; $i <= OBJECTS - 1; $i++) {
    $sum = 0;
    for ($j = 0; $j <= FALLS - 1; $j++) {
      $sum += $g[$i][$j];
    }
    echo "Average g for object No ", $i + 1, ": ", $sum / FALLS;
  }

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

  $sum = 0;
  for ($i = 0; $i <= OBJECTS - 1; $i++) {
    for ($j = 0; $j <= FALLS - 1; $j++) {
      $sum += $g[$i][$j];
    }
  }
  echo "Overall average g: ", $sum / (OBJECTS * FALLS);
?>
```

3. Solution

```
<?php
  define("PLAYERS", 15);
  define("MATCHES", 12);

  $points = array(array());
  for ($i = 0; $i <= PLAYERS - 1; $i++) {
    for ($j = 0; $j <= MATCHES - 1; $j++) {
      $points[$i][$j] = trim(fgets(STDIN));
    }
  }

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

  for ($j = 0; $j <= MATCHES - 1; $j++) {
    $sum = 0;
    for ($i = 0; $i <= PLAYERS - 1; $i++) {
      $sum += $points[$i][$j];
    }
    echo "Total number of points for match No ", $j + 1, ": ", $sum;
  }
?>
```

4. Solution

```
<?php
  define("CITIES", 20);
  define("HOURS", 24);

  $temperatures = array(array());
  for ($i = 0; $i <= CITIES - 1; $i++) {
    for ($j = 0; $j <= HOURS - 1; $j++) {
      $temperatures[$i][$j] = trim(fgets(STDIN));
    }
  }

  for ($j = 0; $j <= HOURS - 1; $j++) {
    $sum = 0;
    for ($i = 0; $i <= CITIES - 1; $i++) {
      $sum += $temperatures[$i][$j];
    }
    if ($sum / CITIES < 10) {
      echo "Hour: ", $j + 1;
    }
  }
?>
```

5. Solution

```
<?php
  define("PLAYERS", 24);
  define("MATCHES", 10);

  $names = array();
  $goals = array(array());
  for ($i = 0; $i <= PLAYERS - 1; $i++) {
    $names[$i] = trim(fgets(STDIN));
    for ($j = 0; $j <= MATCHES - 1; $j++) {
      $goals[$i][$j] = trim(fgets(STDIN));
    }
  }

  for ($i = 0; $i <= PLAYERS - 1; $i++) {
    $sum = 0;
    for ($j = 0; $j <= MATCHES - 1; $j++) {
      $sum += $goals[$i][$j];
    }
    echo $names[$i], ": ", $sum / MATCHES;
  }

  for ($j = 0; $j <= MATCHES - 1; $j++) {
    $sum = 0;
    for ($i = 0; $i <= PLAYERS - 1; $i++) {
      $sum += $goals[$i][$j];
    }
    echo "Match No ", $j + 1, ": ", $sum;
  }
?>
```

6. Solution

```php
<?php
  define("STUDENTS", 24);
  define("LESSONS", 10);

  $names = array();
  $grades = array(array());
  for ($i = 0; $i <= STUDENTS - 1; $i++) {
    $names[$i] = trim(fgets(STDIN));
    for ($j = 0; $j <= LESSONS - 1; $j++) {
      $grades[$i][$j] = trim(fgets(STDIN));
    }
  }

  $average = array();
  for ($i = 0; $i <= STUDENTS - 1; $i++) {
    $sum = 0;
    for ($j = 0; $j <= LESSONS - 1; $j++) {
      $sum += $grades[$i][$j];
    }
    $average[$i] = $sum / LESSONS;
    echo $names[$i], ": ", $average[$i];
  }

  for ($j = 0; $j <= LESSONS - 1; $j++) {
    $sum = 0;
    for ($i = 0; $i <= STUDENTS - 1; $i++) {
      $sum += $grades[$i][$j];
    }
    echo $sum / STUDENTS;
  }

  for ($i = 0; $i <= STUDENTS - 1; $i++) {
    if ($average[$i] < 60) {
      echo $names[$i], "\n";
    }
  }

  for ($i = 0; $i <= STUDENTS - 1; $i++) {
    if ($average[$i] > 89) {
      echo $names[$i], " Bravo!\n";
    }
  }
?>
```

7. Solution

```php
<?php
  define("ARTISTS", 15);
  define("JUDGES", 5);

  $judge_names = array();
  for ($j = 0; $j <= JUDGES - 1; $j++) {
    echo "Enter name for judge No ", $j + 1, ": ";
    $judge_names[$j] = trim(fgets(STDIN));
  }

  $artist_names = array();
  $song_titles = array();
  $score = array(array());
  for ($i = 0; $i <= ARTISTS - 1; $i++) {
    echo "Enter name for artist No ", $i + 1, ": ";
    $artist_names[$i] = trim(fgets(STDIN));
    echo "Enter song title for artist ", $artist_names[$i], ": ";
    $song_titles[$i] = trim(fgets(STDIN));
    for ($j = 0; $j <= JUDGES - 1; $j++) {
      echo "Enter score for artist: ", $artist_names[$i], " gotten from judge ", $judge_names[$j], ":";
      $score[$i][$j] = trim(fgets(STDIN));
    }
  }

  for ($i = 0; $i <= ARTISTS - 1; $i++) {
    $sum = 0;
    for ($j = 0; $j <= JUDGES - 1; $j++) {
      $sum += $score[$i][$j];
    }
    echo $artist_names[$i], ", ", $song_titles[$i], ": ", $sum, "\n";
  }

  for ($j = 0; $j <= JUDGES - 1; $j++) {
    $sum = 0;
    for ($i = 0; $i <= ARTISTS - 1; $i++) {
      $sum += $score[$i][$j];
    }
    echo $judge_names[$j], ": ", $sum / ARTISTS , "\n";
  }
?>
```

8. Solution

```php
<?php
```

```
  define("PEOPLE", 30);
  define("MONTHS", 12);

  $weights = array(array());
  $heights = array(array());
  for ($i = 0; $i <= PEOPLE - 1; $i++) {
    for ($j = 0; $j <= MONTHS - 1; $j++) {
      $weights[$i][$j] = trim(fgets(STDIN));
      $heights[$i][$j] = trim(fgets(STDIN));
    }
  }

  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 / MONTHS;
    $average_height = $sum_heights / MONTHS;
    echo $average_weight, $average_height, $average_weight * 702 / pow($average_height, 2);
  }

  for ($i = 0; $i <= PEOPLE - 1; $i++) {
    echo $weights[$i][4] * 702 / pow($heights[$i][4], 2);
    echo $weights[$i][7] * 702 / pow($heights[$i][7], 2);
  }
?>
```

9. Solution

```
<?php
  define("VAT", 0.19);
  define("CONSUMERS", 1000);

  $meter_reading = array(array());
  for ($i = 0; $i <= CONSUMERS - 1; $i++) {
    $meter_reading[$i][0] = trim(fgets(STDIN));
    $meter_reading[$i][1] = trim(fgets(STDIN));
  }

  $sum = 0;
  for ($i = 0; $i <= CONSUMERS - 1; $i++) {
    $consumed = $meter_reading[$i][1] - $meter_reading[$i][0];
    echo $consumed;
    $payment = $consumed * 0.07;
    $payment += VAT * $payment;
    echo $payment;

    $sum += $consumed;
  }

  echo $sum, $sum * 0.07 + $sum * 0.07 * VAT;
?>
```

10. Solution

```
<?php
  define("CURRENCIES", 4);
  define("DAYS", 5);

  echo "Enter an amount in US dollars: ";
  $usd = trim(fgets(STDIN));

  $currency = array("British Pound Sterling", "Euro", "Canadian Dollar", "Australian Dollar");

  $rate = array(array(1.579, 1.577, 1.572, 1.580, 1.584),
                array(1.269, 1.270, 1.265, 1.240, 1.255),
                array(0.895, 0.899, 0.884, 0.888, 0.863),
                array(0.811, 0.815, 0.822, 0.829, 0.819));

  for ($i = 0; $i <= CURRENCIES - 1; $i++) {
    $sum = 0;
    for ($j = 0; $j <= DAYS - 1; $j++) {
      $sum += $rate[$i][$j];
    }
    $average = $sum / DAYS;
    echo $usd, " US dollars = ", $usd / $average, " ", $currency[$i], "s\n";
  }
?>
```

11. Solution

```
<?php
  define("EMPLOYEES", 10);
  define("DAYS", 5);

  $days = array("Monday", "Tuesday", "Wednesday", "Thursday", "Friday");

  $pay_rate = trim(fgets(STDIN));

  $names = array();
  $hours_worked_per_day = array(array());
```

```
for ($i = 0; $i <= EMPLOYEES - 1; $i++) {
  $names[$i] = trim(fgets(STDIN));
  for ($j = 0; $j <= DAYS - 1; $j++) {
    $hours_worked_per_day[$i][$j] = trim(fgets(STDIN));
  }
}

$hours_worked_per_week = array();
for ($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) {
    echo $names[$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);
  }
  echo $names[$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) {
        echo $names[$i], $days[$j], "Overtime!";
      }
    }
  }
}

for ($j = 0; $j <= DAYS - 1; $j++) {
  $sum = 0;
  for ($i = 0; $i <= EMPLOYEES - 1; $i++) {
    if ($hours_worked_per_day[$i][$j] <= 8) {
      $gross_pay = $pay_rate * $hours_worked_per_day[$i][$j];
    }
    else {
      $gross_pay = $pay_rate * 8 + 1.5 * $pay_rate * ($hours_worked_per_day[$i][$j] - 8);
      }
      $sum += $gross_pay;
    }
    echo $days[$j], $sum;
  }
?>
```

12. Solution

```
<?php
  define("ROWS", 3);
  define("COLUMNS", 4);
  define("ELEMENTS", ROWS * COLUMNS);

  $a = array(array(9, 9, 2, 6),
             array(4, 1, 10, 11),
             array(12, 15, 7, 3));

  $b = array();
  $k = 0;
  for ($i = 0; $i <= ROWS - 1; $i++) {
    for ($j = 0; $j <= COLUMNS - 1; $j++) {
      $b[$k] = $a[$i][$j];
      $k++;
    }
  }

  for ($k = 0; $k <= ELEMENTS - 1; $k++) {
    echo $b[$k], " ";
  }
?>
```

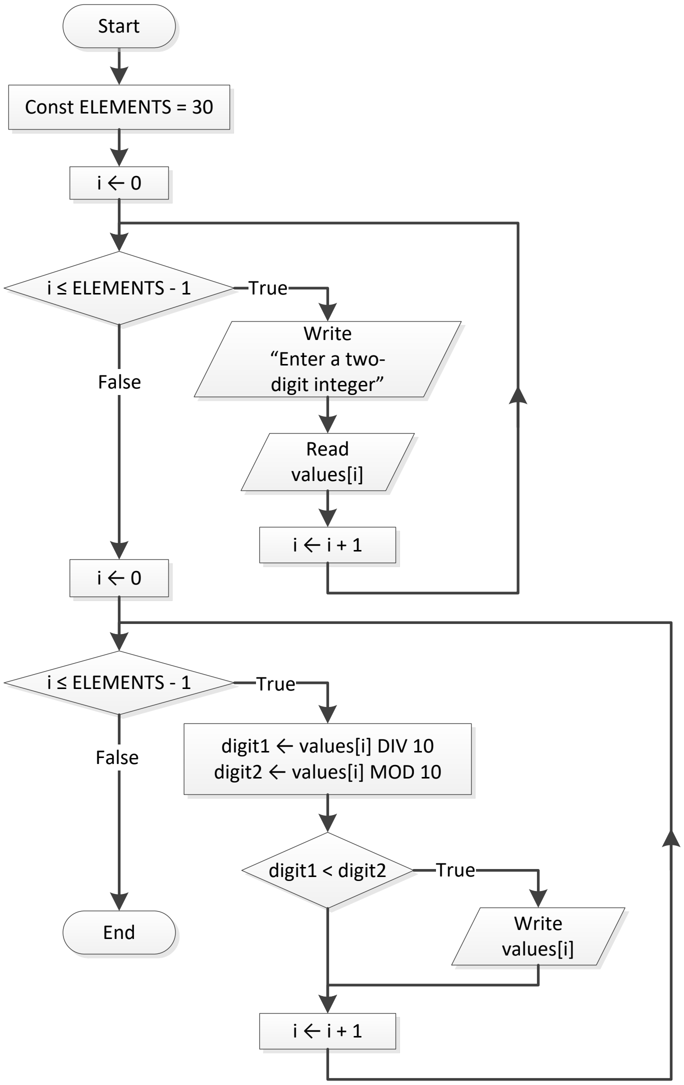
13. Solution

```
<?php
  define("ROWS", 3);
  define("COLUMNS", 3);

  $a = array(16, 12, 3, 5, 6, 9, 18, 19, 20);

  $b = array(array());
  $k = 0;
  for ($i = ROWS - 1; $i >= 0; $i--) {
    for ($j = 0; $j <= COLUMNS - 1; $j++) {
      $b[$i][$j] = $a[$k];
      $k++;
    }
  }

  for ($i = 0; $i <= ROWS - 1; $i++) {
    for ($j = 0; $j <= COLUMNS - 1; $j++) {
      echo $b[$i][$j], "\t";
    }
    echo "\n";
  }
?>
```

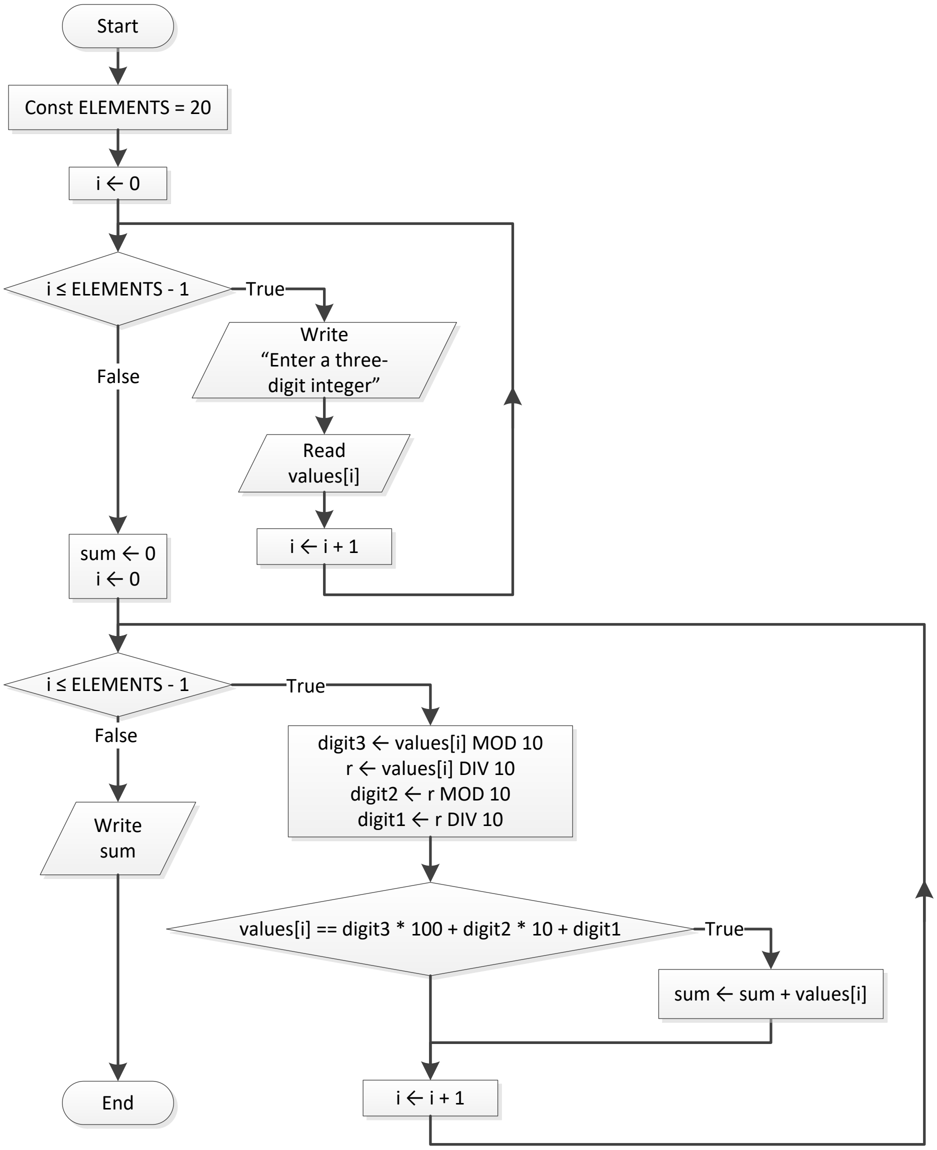
Chapter 36

36.4 Answers of Review Exercises

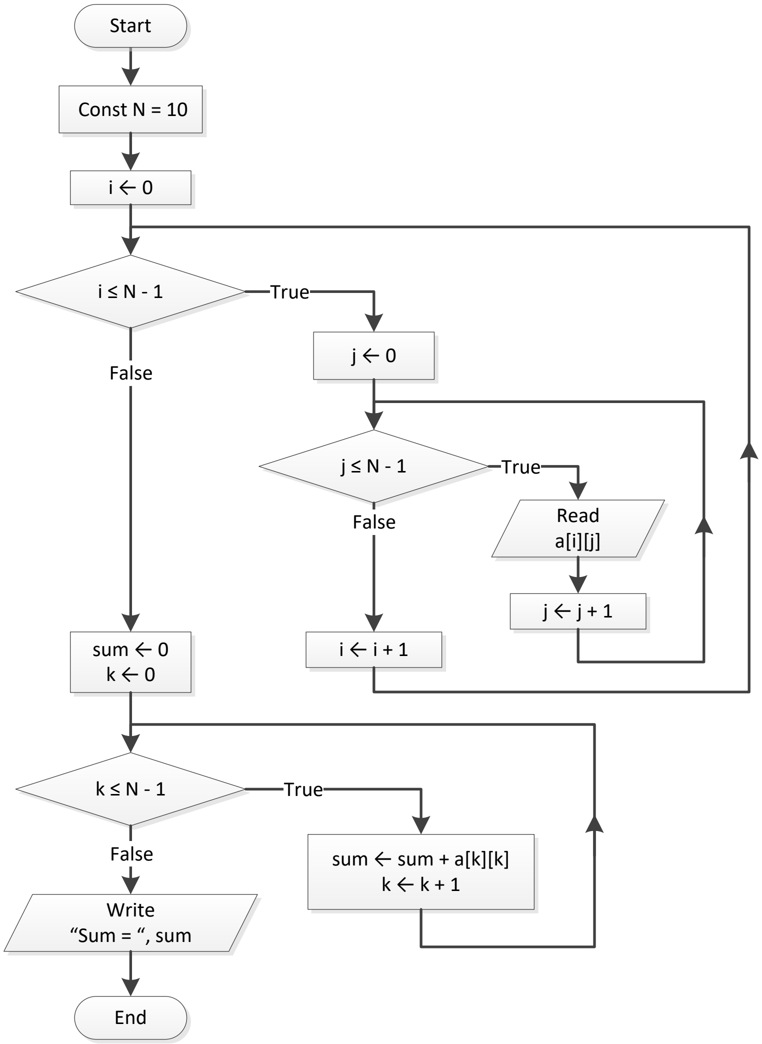
1. Solution



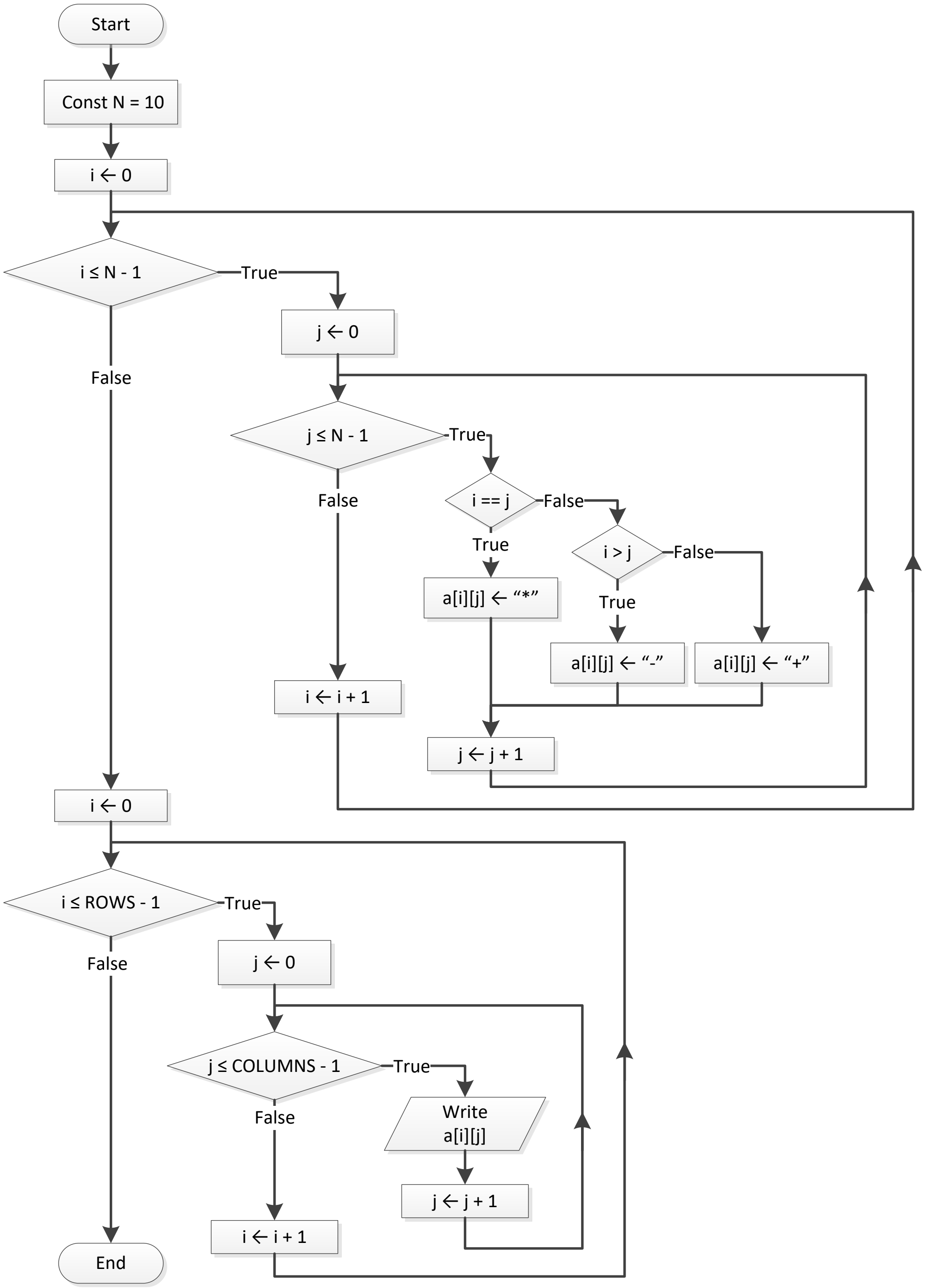
2. Solution


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

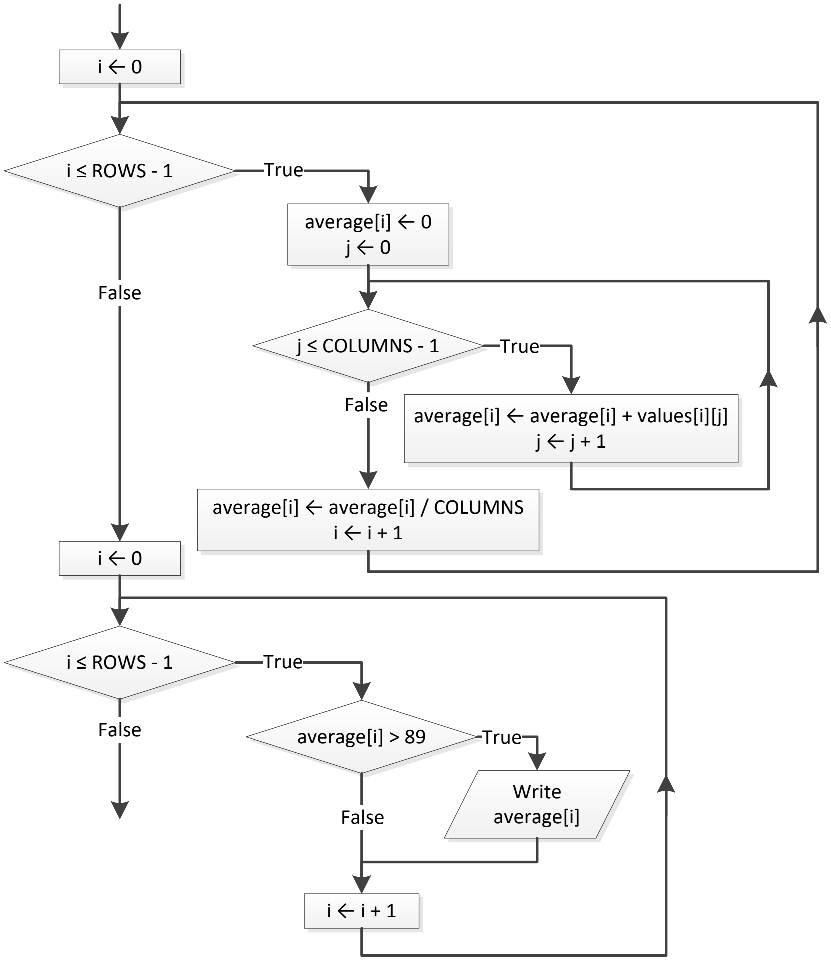
3. Solution


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

4. Solution


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

5. Solution



6. Solution

```
for ($i = 0; $i <= PEOPLE - 1; $i++) {
  do {
    $a[$i] = trim(fgets(STDIN));
  } while (a[i] % 2 == 2);
}
```

7. Solution

```
for ($i = 0; $i <= ELEMENTS - 1; $i++) {
  $a[$i] = trim(fgets(STDIN));
  while (a[i] < 0) {
    echo "Error";
    $a[$i] = trim(fgets(STDIN));
  }
}
```

8. Solution

```
$i = 0;
$S = 0;
$a[i] = trim(fgets(STDIN));
$i++;
while ($i < 90) {
  $S += $a[$i - 1] * $i;
  $a[i] = trim(fgets(STDIN));
  $i++;
}
echo $S;
while ($i >= 0) {
  echo $a[$i];
  $i -= 5;
}
```

9. Solution

```
for ($i = 0; $i <= ROWS - 1; $i++) {
  $max = $a[$i][0];
  for ($j = 1; $j <= COLUMNS - 1; $j++) {
    if ($a[$i][$j] > $max) {
      $max = $a[$i][$j];
    }
  }
  echo $max;
}
```

10. Solution

```
for ($i = 0; $i <= ROWS - 1; $i++) {
  for ($j = 0; $j <= COLUMNS - 1; $j++) {
    $a[$i][$j] = trim(fgets(STDIN));
    while ($a[$i][$j] == 0) {
      echo "Error";
      $a[$i][$j] = trim(fgets(STDIN));
    }
  }
}
```

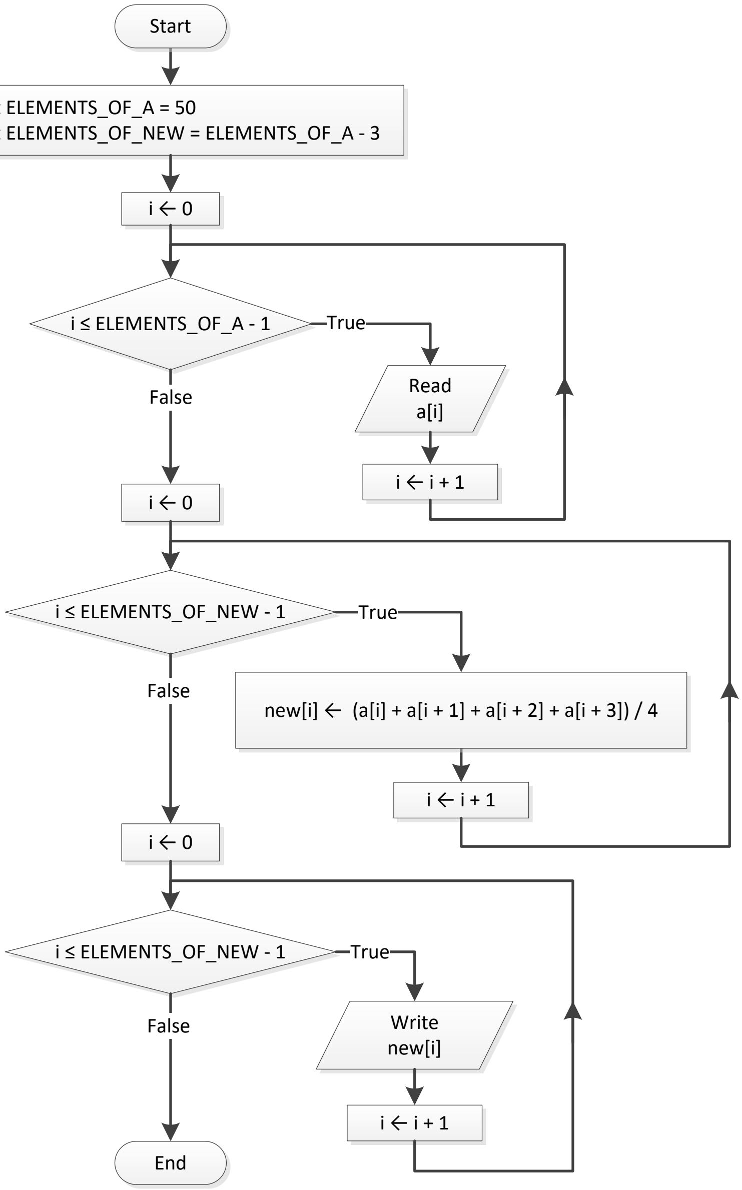
Chapter 37

37.7 Answers of 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. true
28. false
29. true
30. true
31. true
32. false
33. true
34. true
35. true
36. true
37. false
38. true
39. true
40. true

37.8 Answers of Review Exercises

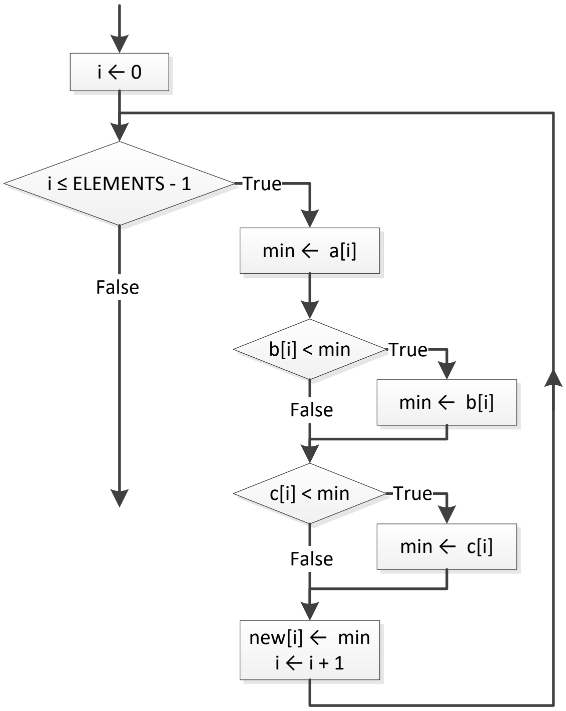
1. Solution



```
<?php
  define("ELEMENTS_OF_A", 50);
  define("ELEMENTS_OF_NEW", ELEMENTS_OF_A - 3);

  $a = array();
  for ($i = 0; $i <= ELEMENTS_OF_A - 1; $i++) {
    $a[$i] = trim(fgets(STDIN));
  }

  $new = array();
  for ($i = 0; $i <= ELEMENTS_OF_NEW - 1; $i++) {
    $new[$i] = ($a[$i] + $a[$i + 1] + $a[$i + 2] + $a[$i + 3]) / 4;
  }

  for ($i = 0; $i <= ELEMENTS_OF_NEW - 1; $i++) {
    echo $new[$i], "\t";
  }
?>
```

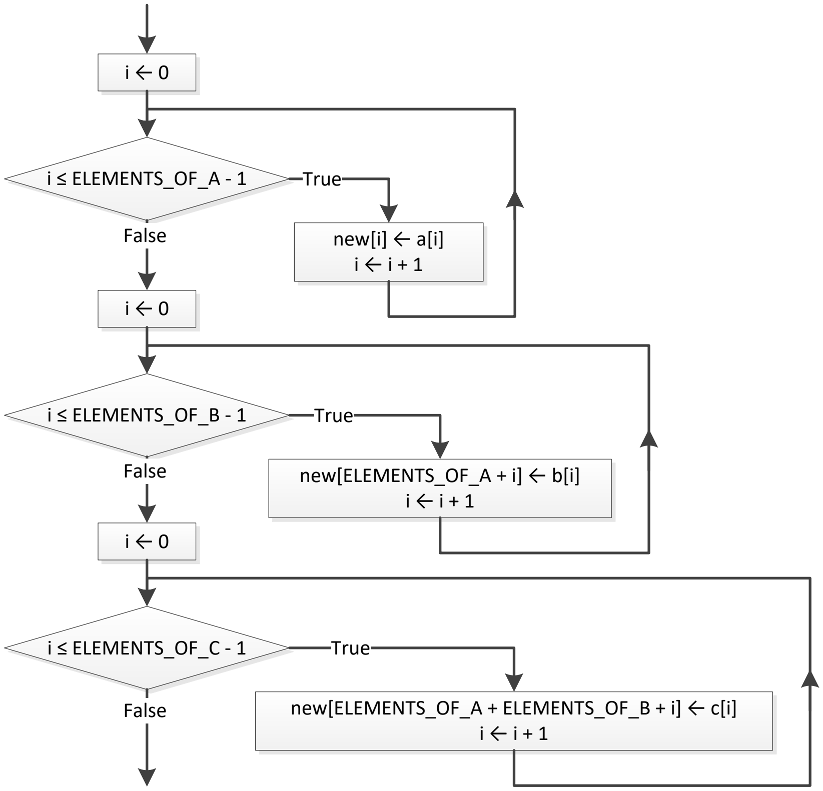
2. Solution

```
<?php
  define("ELEMENTS", 15);

  $a = array();
  for ($i = 0; $i <= ELEMENTS - 1; $i++) {
    $a[$i] = trim(fgets(STDIN));
  }
  $b = array();
  for ($i = 0; $i <= ELEMENTS - 1; $i++) {
    $b[$i] = trim(fgets(STDIN));
  }

  $c = array();
  for ($i = 0; $i <= ELEMENTS - 1; $i++) {
    $c[$i] = trim(fgets(STDIN));
  }

  $new = array();
  for ($i = 0; $i <= ELEMENTS - 1; $i++) {
    $min = $a[$i];
    if ($b[$i] < $min) {
      $min = $b[$i];
    }
    if ($c[$i] < $min) {
      $min = $c[$i];
    }
    $new[$i] = $min;
  }

  for ($i = 0; $i <= ELEMENTS - 1; $i++) {
    echo $new[$i], "\n";
  }
?>
```



3. Solution

```php
<?php
  define("ELEMENTS_OF_A", 10);
  define("ELEMENTS_OF_B", 5);
  define("ELEMENTS_OF_C", 15);
  define("ELEMENTS_OF_NEW", ELEMENTS_OF_A + ELEMENTS_OF_B + ELEMENTS_OF_C);

  $a = array();
  for ($i = 0; $i <= ELEMENTS_OF_A - 1; $i++) {
    $a[$i] = trim(fgets(STDIN));
  }
  $b = array();
  for ($i = 0; $i <= ELEMENTS_OF_B - 1; $i++) {
    $b[$i] = trim(fgets(STDIN));
  }
  $c = array();
  for ($i = 0; $i <= ELEMENTS_OF_C - 1; $i++) {
    $c[$i] = trim(fgets(STDIN));
  }

  $new = array();
  for ($i = 0; $i <= ELEMENTS_OF_C - 1; $i++) {
    $new[$i] = $c[$i];
  }
  for ($i = 0; $i <= ELEMENTS_OF_B - 1; $i++) {
    $new[ELEMENTS_OF_C + $i] = $b[$i];
  }
  for ($i = 0; $i <= ELEMENTS_OF_A - 1; $i++) {
    $new[ELEMENTS_OF_B + ELEMENTS_OF_C + $i] = $a[$i];
  }

  //Display array $new
  for ($i = 0; $i <= ELEMENTS_OF_NEW - 1; $i++) {
    echo $new[$i], "\t";
  }
?>
```

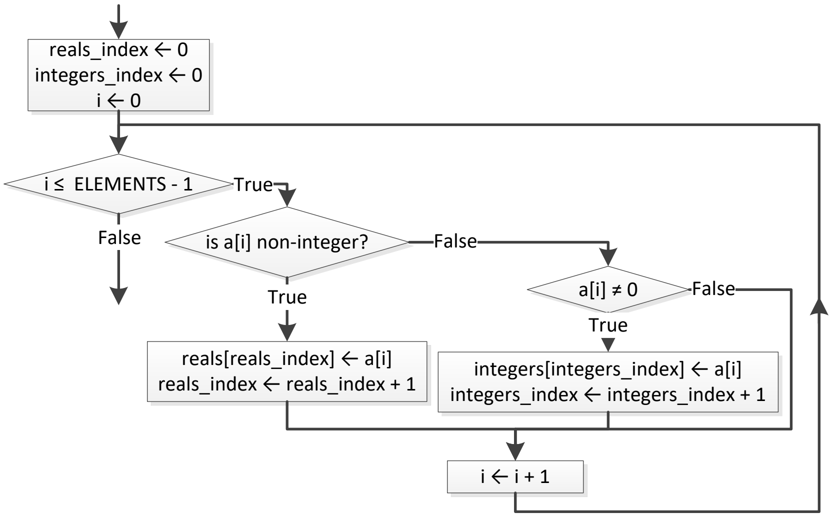


4. Solution

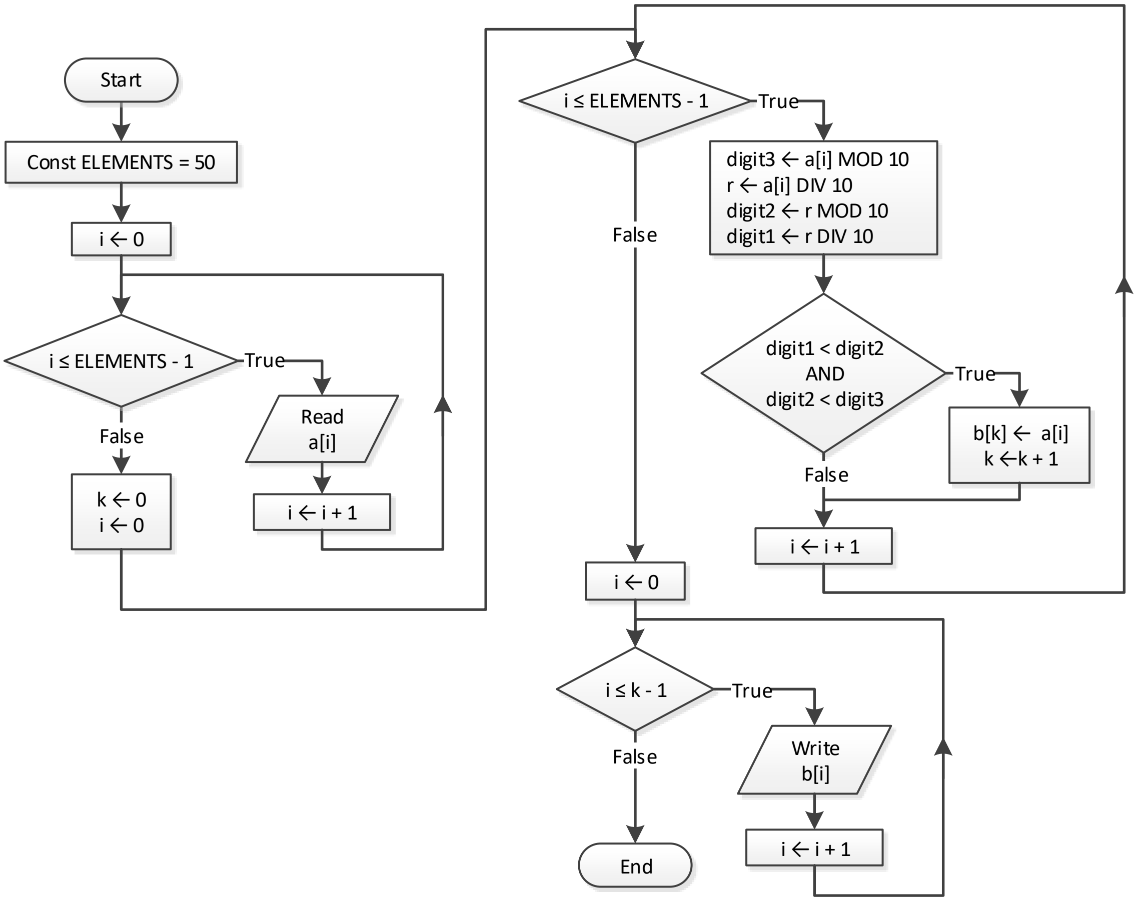
```
<?php
  define("COLUMNS_OF_A", 10);
  define("COLUMNS_OF_B", 15);
  define("COLUMNS_OF_C", 20);
  define("ROWS", 5);
  define("COLUMNS", COLUMNS_OF_A + COLUMNS_OF_B + COLUMNS_OF_C);

  $a = array(array());
  for ($i = 0; $i <= ROWS - 1; $i++) {
    for ($j = 0; $j <= COLUMNS_OF_A - 1; $j++) {
      $a[$i][$j] = trim(fgets(STDIN));
    }
  }

  $b = array(array());
  for ($i = 0; $i <= ROWS - 1; $i++) {
    for ($j = 0; $j <= COLUMNS_OF_B - 1; $j++) {
      $b[$i][$j] = trim(fgets(STDIN));
    }
  }

  $c = array(array());
  for ($i = 0; $i <= ROWS - 1; $i++) {
    for ($j = 0; $j <= COLUMNS_OF_C - 1; $j++) {
      $c[$i][$j] = trim(fgets(STDIN));
    }
  }

  $new = array(array());
  for ($i = 0; $i <= ROWS - 1; $i++) {
    for ($j = 0; $j <= COLUMNS_OF_A - 1; $j++) {
      $new[$i][$j] = $a[$i][$j];
    }
  }
  for ($i = 0; $i <= ROWS - 1; $i++) {
    for ($j = 0; $j <= COLUMNS_OF_B - 1; $j++) {
      $new[$i][COLUMNS_OF_A + $j] = $b[$i][$j];
    }
  }

  for ($i = 0; $i <= ROWS - 1; $i++) {
    for ($j = 0; $j <= COLUMNS_OF_C - 1; $j++) {
      $new[$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++) {
      echo $new[$i][$j], "\t";
    }
    echo "\n";
  }
?>
```

5. Solution

```
<?php
  define("ELEMENTS", 50);

  $a = array();
  for ($i = 0; $i <= ELEMENTS - 1; $i++) {
    $a[$i] = trim(fgets(STDIN));
  }

  $reals = array();
  $integers = array();
  $reals_index = 0;
  $integers_index = 0;
  for ($i = 0; $i <= ELEMENTS - 1; $i++) {
    if ($a[$i] != intval($a[$i])) {
      $reals[$reals_index] = $a[$i];
      $reals_index++;
    }
    elseif ($a[$i] != 0) {
      $integers[$integers_index] = $a[$i];
      $integers_index++;
    }
  }

  for ($i = 0; $i <= $reals_index - 1; $i++) {
    echo $reals[$i], "\t";
  }

  echo "\n";
  for ($i = 0; $i <= $integers_index - 1; $i++) {
    echo $integers[$i], "\t";
  }
?>
```


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

6. Solution



```
<?php
  define("ELEMENTS", 50);
  $a = array();
  for ($i = 0; $i <= ELEMENTS - 1; $i++) {
    $a[$i] = trim(fgets(STDIN));
  }

  $b = array();
  $k = 0;
  for ($i = 0; $i <= ELEMENTS - 1; $i++) {
    $digit3 = $a[$i] % 10;
    $r = intval($a[$i] / 10);
    $digit2 = $r % 10;
    $digit1 = intval($r / 10);

    if ($digit1 < $digit2 && $digit2 < $digit3) {
      $b[$k] = $a[$i];
      $k++;
    }
  }

  for ($i = 0; $i <= $k - 1; $i++) {
    echo $b[$i], "\t";
  }
?>
```

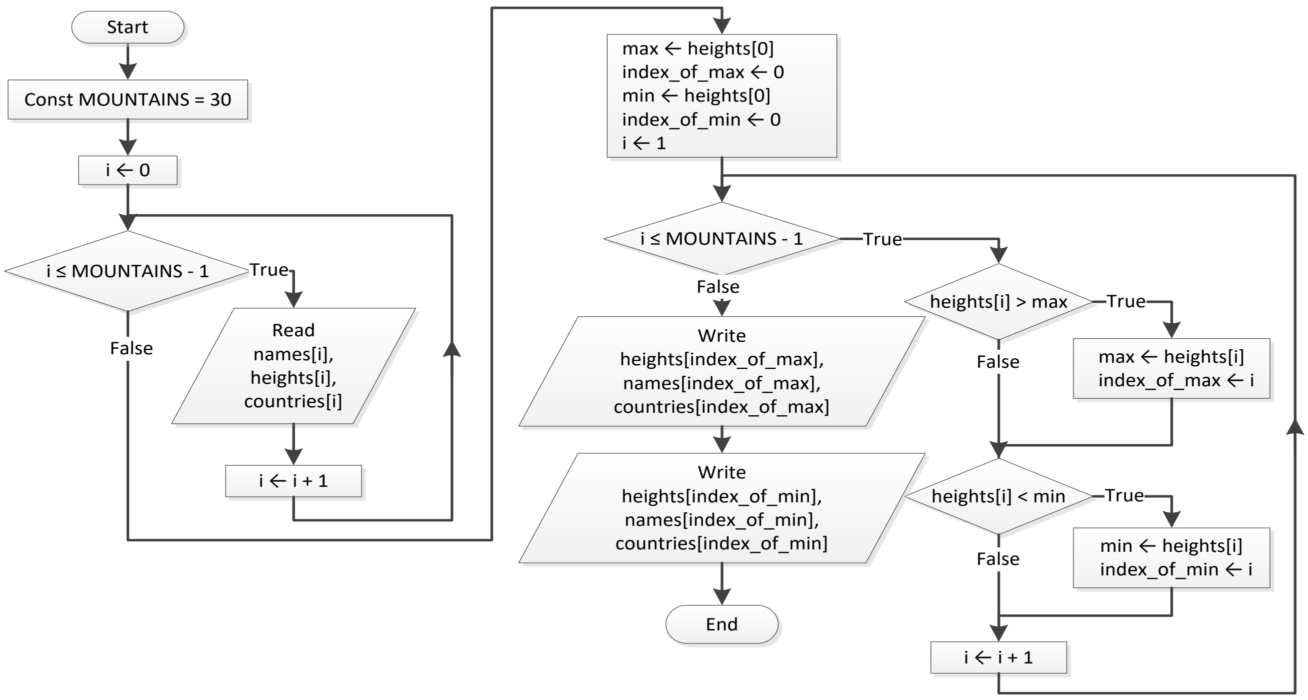
7. Solution

```
<?php
  define("PRODUCTS", 10);
  define("CITIZENS", 1000);

  $prod_names = array();
  $answers = array(array());
  for ($i = 0; $i <= PRODUCTS - 1; $i++) {
    $prod_names[$i] = trim(fgets(STDIN));
    for ($j = 0; $j <= CITIZENS - 1; $j++) {
      $answers[$i][$j] = trim(fgets(STDIN));
      while ($answers[$i][$j] < "A" || $answers[$i][$j] > "D") {
        echo "Error! ";
        $answers[$i][$j] = trim(fgets(STDIN));
      }
    }
  }

  $count_A = array();
  for ($i = 0; $i <= PRODUCTS - 1; $i++) {
    $count_A[$i] = 0;
    for ($j = 0; $j <= CITIZENS - 1; $j++) {
      if ($answers[$i][$j] == "A") {
        $count_A[$i]++;
      }
    }
    echo $prod_names[$i], $count_A[$i];
  }

  for ($j = 0; $j <= CITIZENS - 1; $j++) {
    $count_B = 0;
    for ($i = 0; $i <= PRODUCTS - 1; $i++) {
      if ($answers[$i][$j] == "B") {
        $count_B++;
      }
    }
    echo $count_B;
  }

  $max = $count_A[0];
  for ($i = 1; $i <= PRODUCTS - 1; $i++) {
    if ($count_A[$i] > $max) {
      $max = $count_A[$i];
    }
  }
  for ($i = 0; $i <= PRODUCTS - 1; $i++) {
    if ($count_A[$i] == $max) {
      echo $prod_names[$i];
    }
  }
?>
```

8. Solution

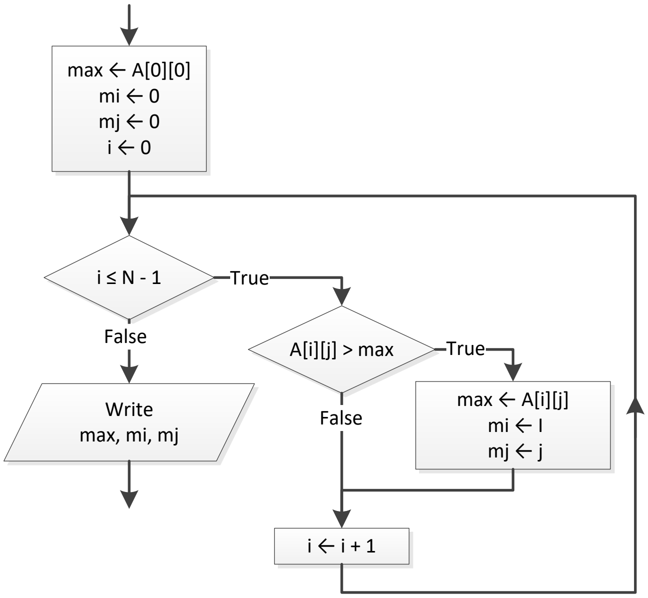
```
<?php
  define("US_CITIES", 20);
  define("CANADIAN_CITIES", 20);

  $us_names = array();
  for ($i = 0; $i <= US_CITIES - 1; $i++) {
    echo "Enter name for US city No ", $i + 1, ": ";
    $us_names[$i] = trim(fgets(STDIN));
  }

  $canadian_names = array();
  for ($j = 0; $j <= CANADIAN_CITIES - 1; $j++) {
    echo "Enter name for Canadian city No ", $j + 1, ": ";
    $canadian_names[$j] = trim(fgets(STDIN));
  }
  $distances = array(array());
  for ($i = 0; $i <= US_CITIES - 1; $i++) {
    for ($j = 0; $j <= CANADIAN_CITIES - 1; $j++) {
      echo "Enter distance between ", $us_names[$i], " and ", $canadian_names[$j], ": ";
      $distances[$i][$j] = trim(fgets(STDIN));
    }
  }

  for ($i = 0; $i <= US_CITIES - 1; $i++) {
    $min = $distances[$i][0];
    $min_j = 0;
    for ($j = 1; $j <= CANADIAN_CITIES - 1; $j++) {
      if ($distances[$i][$j] < $min) {
        $min = $distances[$i][$j];
        $min_j = $j;
      }
    }
    echo "Closest Canadian city to ", $us_names[$i], " is ", $canadian_names[$min_j], "\n";
  }


?>
```

9. Solution



```php
<?php
  define("MOUNTAINS", 30);

  $names = array();
  $heights = array();
  $countries = array();
  for ($i = 0; $i <= MOUNTAINS - 1; $i++) {
    $names[$i] = trim(fgets(STDIN));
    $heights[$i] = trim(fgets(STDIN));
    $countries[$i] = trim(fgets(STDIN));
  }

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

  echo $heights[$index_of_max], $names[$index_of_max], $countries[$index_of_max];
  echo $heights[$index_of_min], $names[$index_of_min], $countries[$index_of_min];
?>
```

10. Solution



11. Solution

```
<?php
  define("TEAMS", 26);
  define("GAMES", 15);

  $names = array();
  $results = array(array());
  for ($i = 0; $i <= TEAMS - 1; $i++) {
    $names[$i] = trim(fgets(STDIN));
    for ($j = 0; $j <= GAMES - 1; $j++) {
      $results[$i][$j] = trim(fgets(STDIN));
    }
  }

  $points = array();
  for ($i = 0; $i <= TEAMS - 1; $i++) {
    $points[$i] = 0;
    for ($j = 0; $j <= GAMES - 1; $j++) {
      if ($results[$i][$j] == "W") {
        $points[$i] += 3;
      }
      elseif ($results[$i][$j] == "T") {
        $points[$i] += 1;
      }
    }
  }

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

  echo $names[$m_i];
?>
```

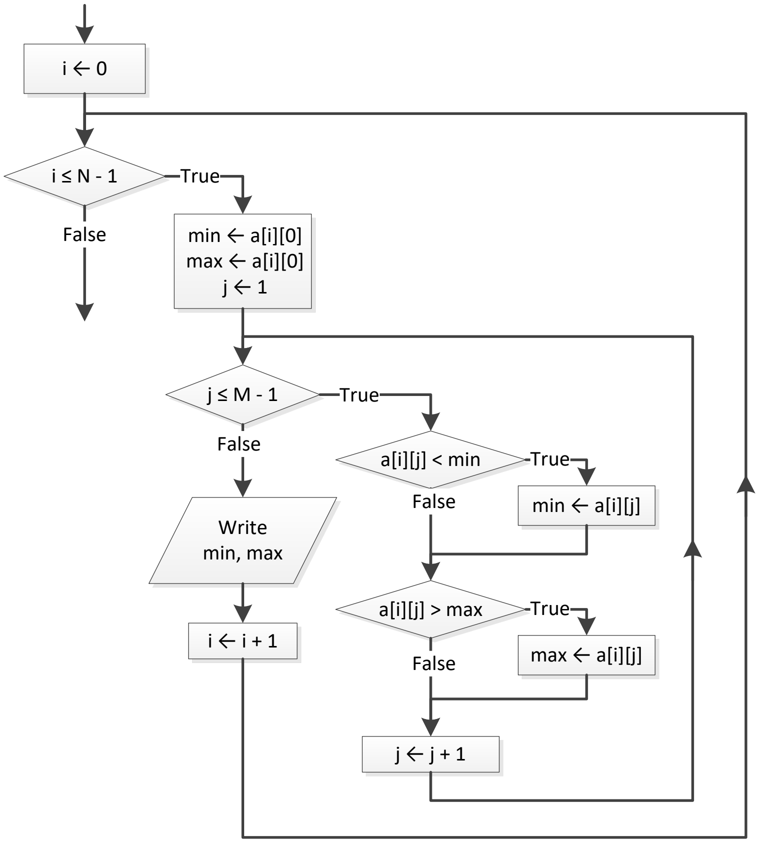
12. Solution

```
<?php
  define("OBJECTS", 10);
  define("FALLS", 20);

  $heights = array(array());
  $times = array(array());
  for ($i = 0; $i <= OBJECTS - 1; $i++) {
    for ($j = 0; $j <= FALLS - 1; $j++) {
      $heights[$i][$j] = trim(fgets(STDIN));
      $times[$i][$j] = trim(fgets(STDIN));
    }
  }

  $g = array(array());
  for ($i = 0; $i <= OBJECTS - 1; $i++) {
    for ($j = 0; $j <= FALLS - 1; $j++) {
      $g[$i][$j] = 2 * $heights[$i][$j] / pow($times[$i][$j], 2);
    }
  }

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

  for ($i = 0; $i <= OBJECTS - 1; $i++) {
    echo $min[$i], $max[$i];
  }

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

  echo $minimum, $maximum;
?>
```

13. Solution

```
<?php
  define("STATIONS", 10);
  define("DAYS", 365);

  $names = array();
  $co2 = array(array());
  for ($i = 0; $i <= STATIONS - 1; $i++) {
    $names[$i] = trim(fgets(STDIN));
    for ($j = 0; $j <= DAYS - 1; $j++) {
      $co2[$i][$j] = trim(fgets(STDIN));
    }
  }

  $average = array();
  for ($i = 0; $i <= STATIONS - 1; $i++) {
    $average[$i] = 0;
    for ($j = 0; $j <= DAYS - 1; $j++) {
      $average[$i] += $co2[$i][$j];
    }
    $average[$i] /= DAYS;
  }

  $min = $average[0];
  $m_i = 0;
  for ($i = 1; $i <= STATIONS - 1; $i++) {
    if ($average[$i] < $min) {
      $min = $average[$i];
      $m_i = $i;
    }
  }
  echo $names[$m_i];
?>
```

14. Solution



15. Solution

```
<?php
  define("TEAMS", 20);
  define("GAMES", 10);

  $names = array();
  $results = array(array());
  for ($i = 0; $i <= TEAMS - 1; $i++) {
    echo "Enter team name: ";
    $names[$i] = trim(fgets(STDIN));
    for ($j = 0; $j <= GAMES - 1; $j++) {
      echo "Enter result for team ", $names[$i], " for game No ", $j + 1, ": ";
      $results[$i][$j] = trim(fgets(STDIN));
      while ($results[$i][$j] != "W" && $results[$i][$j] != "L" && $results[$i][$j] != "T") {
        echo "Error! Enter only value W, L, or T: ";
        $results[$i][$j] = trim(fgets(STDIN));
      }
    }
  }

  $points = array();
  for ($i = 0; $i <= TEAMS - 1; $i++) {
    $points[$i] = 0;
    for ($j = 0; $j <= GAMES - 1; $j++) {
      if ($results[$i][$j] == "W") {
        $points[$i] += 3;
      }
      elseif ($results[$i][$j] == "T") {
        $points[$i] += 1;
      }
    }
  }

  for ($m = 1; $m <= TEAMS - 1; $m++) {
    $swaps = false;
    for ($n = TEAMS - 1; $n >= $m; $n--) {
      if ($points[$n] > $points[$n - 1]) {
        $temp = $points[$n];
        $points[$n] = $points[$n - 1];
        $points[$n - 1] = $temp;

        $temp = $names[$n];
        $names[$n] = $names[$n - 1];
        $names[$n - 1] = $temp;

        $swaps = true;
      }
    }
    if ($swaps == false) break;
  }

  echo "Gold: ", $names[0], "\n";
  echo "Silver: ", $names[1], "\n";
  echo "Bronze: ", $names[2], "\n";
?>
```

16. Solution

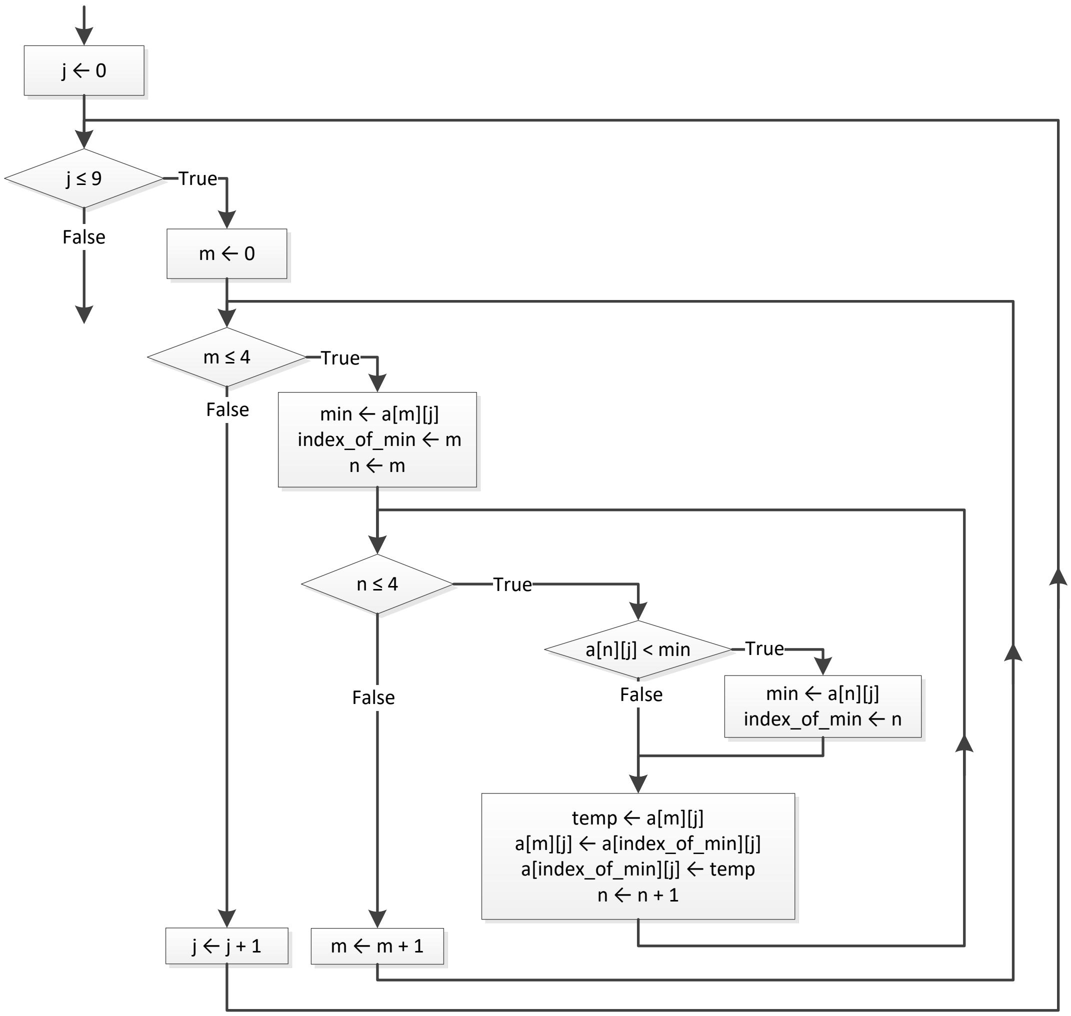
```php
<?php
  define("PEOPLE", 50);

  $names = array();
  $heights = array();
  for ($i = 0; $i <= PEOPLE - 1; $i++) {
    echo "Enter name for person No. ", $i + 1, ": ";
    $names[$i] = trim(fgets(STDIN));
    echo "Enter height for person No. ", $i + 1, ": ";
    $heights[$i] = trim(fgets(STDIN));
  }

  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 = $names[$n];
        $names[$n] = $names[$n - 1];
        $names[$n - 1] = $temp;
      }
      elseif ($heights[$n] == $heights[$n - 1]) {
        if ($names[$n] < $names[$n - 1]) {
          $temp = $names[$n];
          $names[$n] = $names[$n - 1];
          $names[$n - 1] = $temp;
        }
      }
    }
  }

  for ($i = 0; $i <= PEOPLE - 1; $i++) {
    echo $heights[$i], "\t", $names[$i], "\n";
  }
?>
```

17. Solution

```php
<?php
  define("ARTISTS", 12);
  define("JUDGES", 10);

```

```
$artist_names = array();
$score = array(array());
for ($i = 0; $i <= ARTISTS - 1; $i++) {
  echo "Enter name for artist No ", $i + 1, ": ";
  $artist_names[$i] = trim(fgets(STDIN));
  for ($j = 0; $j <= JUDGES - 1; $j++) {
    echo "Enter score for artist: ", $artist_names[$i], " gotten from judge No ", $j + 1, ": ";
    $score[$i][$j] = trim(fgets(STDIN));
  }
}

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

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

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

      $temp = $artist_names[$n];
      $artist_names[$n] = $artist_names[$n - 1];
      $artist_names[$n - 1] = $temp;
    }
    elseif ($sum[$n] == $sum[$n - 1]) {
        if ($artist_names[$n] < $artist_names[$n - 1]) {
          $temp = $artist_names[$n];
          $artist_names[$n] = $artist_names[$n - 1];
          $artist_names[$n - 1] = $temp;
        }
      }
    }
  }

  for ($i = 0; $i <= ARTISTS - 1; $i++) {
    echo $artist_names[$i], $sum[$i];
  }
?>
```

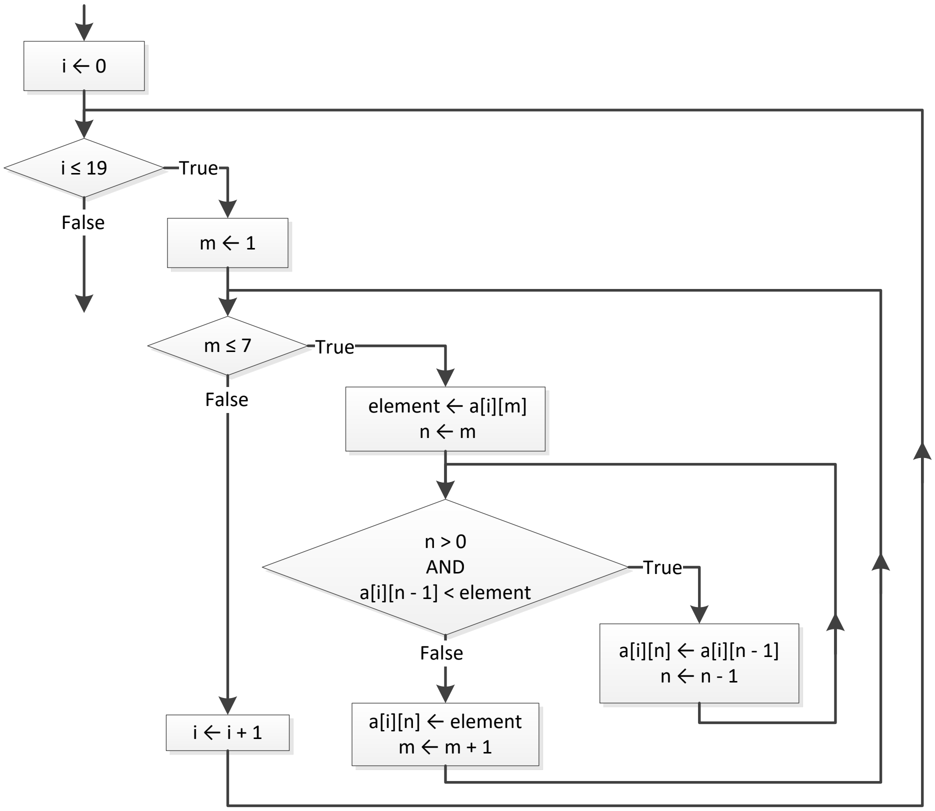
18. Solution



19. Solution

```
<?php
  define("PEOPLE", 10);
  define("PUZZLES", 8);

  $names = array();
  $times = array(array());
  for ($i = 0; $i <= PEOPLE - 1; $i++) {
    $names[$i] = trim(fgets(STDIN));
    for ($j = 0; $j <= PUZZLES - 1; $j++) {
      $times[$i][$j] = trim(fgets(STDIN));
    }
  }

  for ($i = 0; $i <= PEOPLE - 1; $i++) {
    for ($m = 0; $m <= PUZZLES - 1; $m++) {
      $min = $times[$i][$m];
      $index_of_min = $m;
      for ($n = $m; $n <= PUZZLES - 1; $n++) {
        if ($times[$i][$n] < $min) {
          $min = $times[$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++) {
    echo $names[$i];
    for ($j = 0; $j <= 2; $j++) {
      echo $times[$i][$j];
    }
  }

  $average = array();
  for ($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++) {
    $min = $average[$m];
    $index_of_min = $m;
    for ($n = $m; $n <= PEOPLE - 1; $n++) {
      if ($average[$n] < $min) {
        $min = $average[$n];
        $index_of_min = $n;
      }
    }
    $temp = $average[$m];
    $average[$m] = $average[$index_of_min];
    $average[$index_of_min] = $temp;

    $temp = $names[$m];
    $names[$m] = $names[$index_of_min];
    $names[$index_of_min] = $temp;
  }

  echo $names[0],$names[1],$names[2];
?>
```

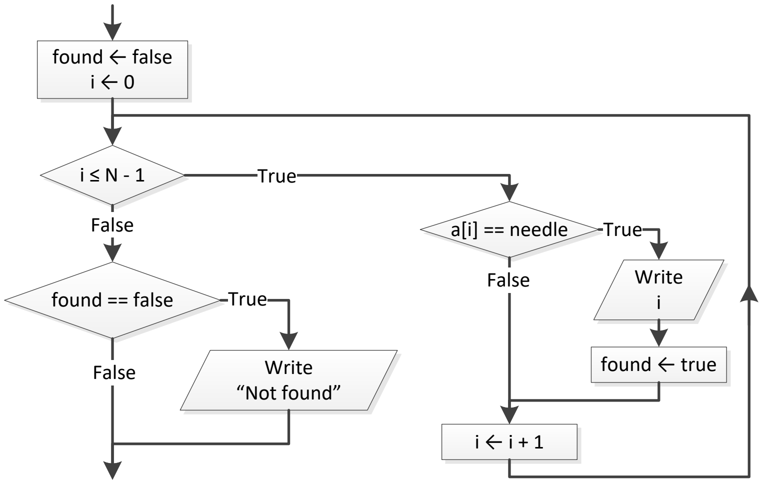
20. Solution



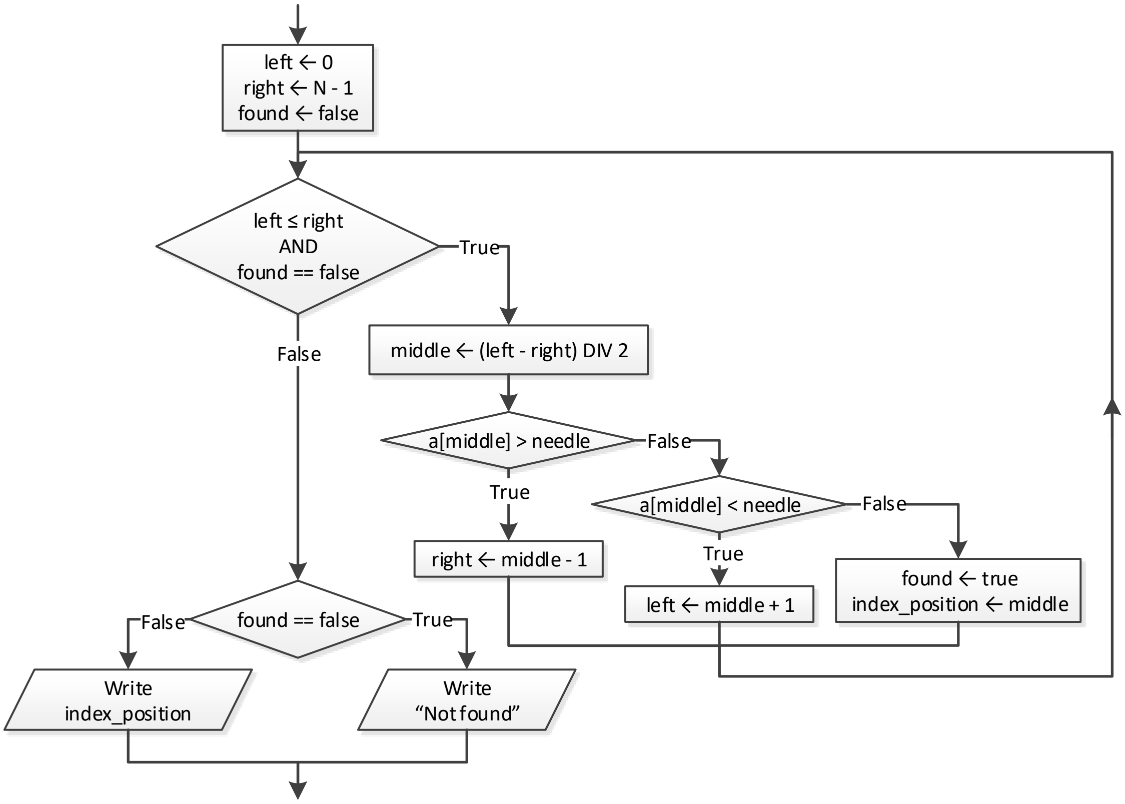
21. Solution

```php
<?php
  define("CITIES", 5);
  define("HOURS", 48);

  $names = array();
  $CO2 = array(array());
  for ($i = 0; $i <= CITIES - 1; $i++) {
    $names[$i] = trim(fgets(STDIN));
    for ($j = 0; $j <= HOURS - 1; $j++) {
      $CO2[$i][$j] = trim(fgets(STDIN));
    }
  }

  $average_per_hour = array();
  for ($i = 0; $i <= CITIES - 1; $i++) {
    $average_per_hour[$i] = 0;
    for ($j = 0; $j <= HOURS - 1; $j++) {
      $average_per_hour[$i] += $CO2[$i][$j];
    }
    $average_per_hour[$i] /= HOURS;
  }

  for ($i = 0; $i <= CITIES - 1; $i++) {
    echo $names[$i], $average_per_hour[$i];
  }

  $average_per_city = array();
  for ($j = 0; $j <= HOURS - 1; $j++) {
    $average_per_city[$j] = 0;
    for ($i = 0; $i <= CITIES - 1; $i++) {
      $average_per_city[$j] += $CO2[$i][$j];
    }
    $average_per_city[$j] /= CITIES;
  }

  for ($j = 0; $j <= HOURS - 1; $j++) {
    echo $average_per_city[$j];
  }

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

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

  for ($m = 1; $m <= CITIES - 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;
  }

  echo $names[0], $names[1], $names[2];
?>
```

22. Solution



23. Solution



24. Solution

```
<?php
  define("TEAMS", 10);
  define("GAMES", 16);

  $names = array();
  $goals_scored = array(array());
  $goals_let_in = array(array());
  for ($i = 0; $i <= TEAMS - 1; $i++) {
    echo "Enter team name: ";
    $names[$i] = trim(fgets(STDIN));
    for ($j = 0; $j <= GAMES - 1; $j++) {
      echo "Enter goals scored: ";
      $goals_scored[$i][$j] = trim(fgets(STDIN));
      while (is_numeric($goals_scored[$i][$j]) != true || $goals_scored[$i][$j] < 0) {
        echo "Error! Enter goals scored: ";
        $goals_scored[$i][$j] = trim(fgets(STDIN));
      }

      echo "Enter goals let in: ";
      $goals_let_in[$i][$j] = trim(fgets(STDIN));
      while (is_numeric($goals_let_in[$i][$j]) != true || $goals_let_in[$i][$j] < 0) {
        echo "Error! Enter goals let in: ";
        $goals_let_in[$i][$j] = trim(fgets(STDIN));
      }
    }
  }

  echo "Enter a team to search: ";
  $needle = trim(fgets(STDIN));

  $i = 0;
  while ($i < TEAMS - 1 && $names[$i] != $needle) {
    $i++;
  }

  if ($names[$i] != $needle) {
    echo "This team does not exist";
  }
  else {
    $sum = 0;
    for ($j = 0; $j <= GAMES - 1; $j++) {
      if ($goals_scored[$i][$j] > $goals_let_in[$i][$j]) {
        $sum += 3;
      }
      elseif ($goals_scored[$i][$j] == $goals_let_in[$i][$j]) {
        $sum += 1;
      }
    }
```

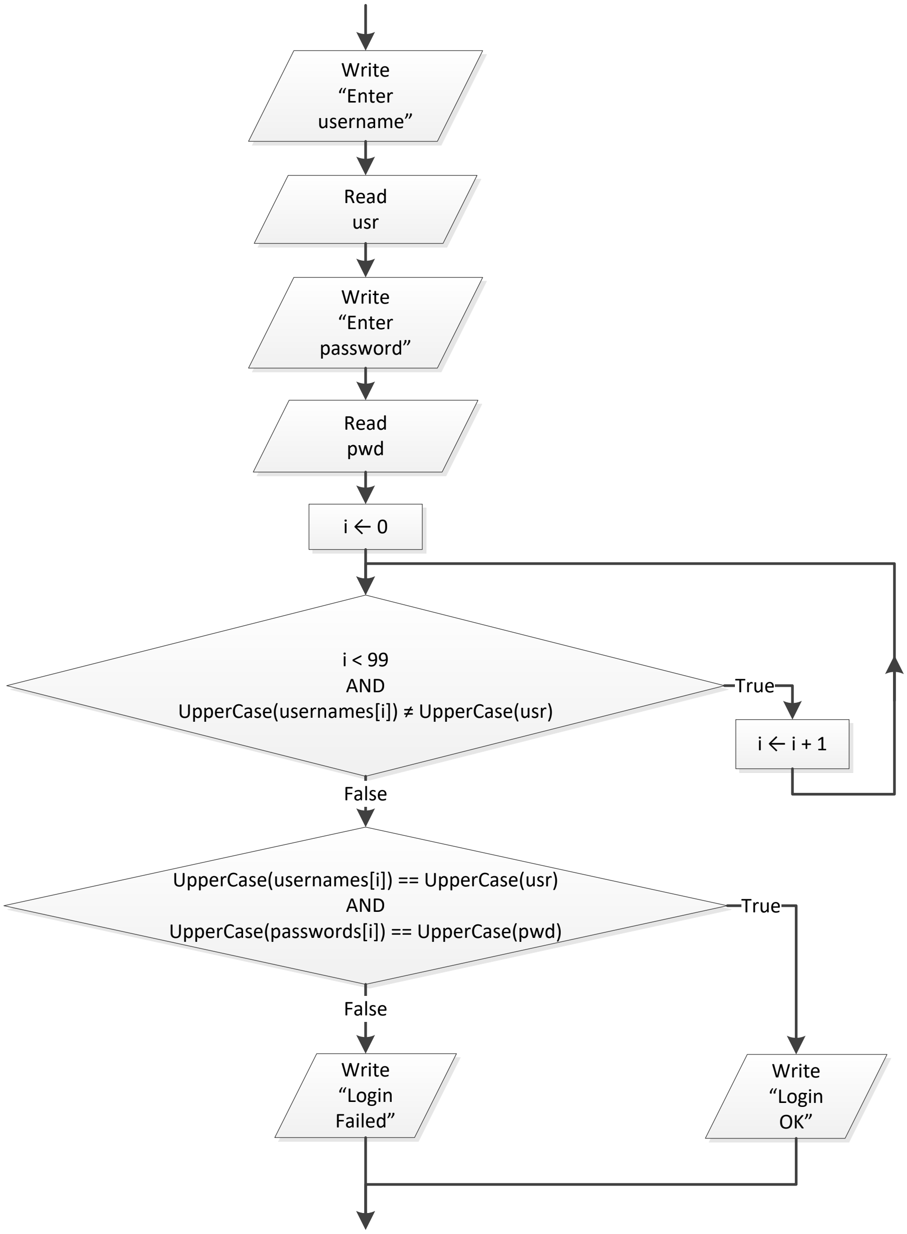
```
    echo $sum;
  }
?>
```

25. Solution

```
<?php
  define("CLASS1", 20);
  define("CLASS2", 25);

  echo "Class A\n";
  $names1 = array();
  for ($i = 0; $i <= CLASS1 - 1; $i++) {
    echo "Enter name: ";
    $names1[$i] = trim(fgets(STDIN));
  }
  echo "Class B\n";
  $names2 = array();
  for ($i = 0; $i <= CLASS2 - 1; $i++) {
    echo "Enter name: ";
    $names2[$i] = trim(fgets(STDIN));
  }

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

  echo "\nClass A\n";
  for ($i = 0; $i <= CLASS1 - 1; $i++) {
    echo $names1[$i], "\n";
  }
```

```
echo "\nClass B\n";
for ($i = 0; $i <= CLASS2 - 1; $i++) {
  echo $names2[$i], "\n";
}

echo "Enter a name to search: ";
$needle = trim(fgets(STDIN));

$left = 0;
$right = CLASS1 - 1;
$found = false;
while ($left <= $right && $found == false) {
  $middle = intval(($left + $right) / 2);

  if ($names1[$middle] > $needle) {
    $right = $middle - 1;
  }
  elseif ($names1[$middle] < $needle) {
    $left = $middle + 1;
  }
  else {
    $found = true;
  }
}

if ($found == true) {
  echo "Student found in class No 1";
}
else {
  $left = 0;
  $right = CLASS2 - 1;
  while ($left <= $right && $found == false) {
    $middle = intval(($left + $right) / 2);

    if ($names2[$middle] > $needle) {
      $right = $middle - 1;
    }
    elseif ($names2[$middle] < $needle) {
      $left = $middle + 1;
    }
    else {
      $found = true;
    }
  }

  if ($found == true) {
      echo "Student found in class No 2";
    }
    else {
      echo "Student not found in either class";
    }
  }
?>
```

26. Solution



Write "Enter username"
Read usr
Write "Enter password"
Read pwd
i ← 0
i < 99 AND UpperCase(usernames[i]) ≠ UpperCase(usr)
True
i ← i + 1
False
UpperCase(usernames[i]) == UpperCase(usr) AND UpperCase(passwords[i]) == UpperCase(pwd)
True
False
Write "Login Failed"
Write "Login OK"

```
echo "Enter username: ";
```

```
$usr = trim(fgets(STDIN));
echo "Enter password: ";
$pwd = trim(fgets(STDIN));

$i = 0;
while ($i < 99 && strtoupper($usernames[$i]) != strtoupper($usr)) {
  $i++;
}

if (strtoupper($usernames[$i]) == strtoupper($usr) && strtoupper($passwords[$i]) == strtoupper($pwd)) {
  echo "Login OK!";
}
else {
  echo "Login Failed!";
}
```

27. Solution

```
echo "Enter a value to search: ";
$value = trim(fgets(STDIN));

$found = false;

//Check if entered value is a valid nine-digit SSN
if (is_numeric($value) && $value >= 100000000 && $value <= 999999999) {
  $i = 0;
  while ($i < 999 && $SSNs[$i] != $value) {
    $i++;
  }

  if ($SSNs[$i] == $value) {
    $found = true;
    echo $names[$i];
  }
}
else {
  for ($i = 0; $i <= 999; $i++) {
    if ($names[$i] == $value) {
      echo $names[$i];
      $found = true;
    }
  }
}

if ($found == false) {
  echo "This value does not exist";
}
```

28. Solution

```php
<?php
  define("STUDENTS", 12);
  define("LESSONS", 6);

  $grades = array(array());
  for ($i = 0; $i <= STUDENTS - 1; $i++) {
    for ($j = 0; $j <= LESSONS - 1; $j++) {
      $grades[$i][$j] = trim(fgets(STDIN));
    }
  }

  $average = array();
  for ($i = 0; $i <= STUDENTS - 1; $i++) {
    $average[$i] = 0;
    for ($j = 0; $j <= LESSONS - 1; $j++) {
      $average[$i] += $grades[$i][$j];
    }
    $average[$i] /= LESSONS;
  }

  $found = false;
  for ($i = 0; $i <= STUDENTS - 1; $i++) {
    if ($average[$i] < 70) {
      $found = true;
    }
  }

  if ($found == true) {
    echo "There is at least one student that has an average value below 70";
  }
?>
```

Chapter 38

38.4 Review Questions: True/False

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

Chapter 39

39.5 Review Questions: True/False

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

39.6 Review Exercises

1. Solution

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

2. Solution

Step	Statement	Main Code		Function sum_digits()		
		$s	$i	$a	$d1	$d2
1	`$s = 0`	**0**	?			
2	`$i = 25`	0	**25**			
3	`$i <= 27`	True				
4	`$s += sum_digits($i)`			**25**	?	?
5	`$d1 = $a % 10`			25	**5**	?
6	`$d2 = intval($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 = intval($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 = intval($a / 10)`			27	7	**2**
19	`return $d1 + $d2`	**24**	27			
20	`$i++`	24	**28**			
21	`$i <= 27`	False				
22	`echo $s`	24 is displayed				

3. Solution

Step	Statement	Main Code		Function sss ()		
		$s	$i	$a	$sum	$k
1	`$i = 1`	?	**1**			
2	`$s = 0`	**0**	1			
3	`while($i < 6)`	True				
4	`if ($i % 2 == 1)`	True				
5	`$s += 1`	**1**	1			
6	`$i++`	1	**2**			
7	`while($i < 6)`	True				
8	`if ($i % 2 == 1)`	False				
9	`$s += sss($i)`			**2**	?	?
10	`$sum = 0`			2	**0**	?
11	`$k = 1`			2	0	**1**
12	`$k <= $a`			True		
13	`$sum += $k`			2	**1**	1
14	`$k++`			2	1	**2**
15	`$k <= $a`			True		
16	`$sum += $k`			2	**3**	2
17	`$k++`			2	3	**3**
18	`$k <= $a`			False		

19	`return $sum`	**4**	2			
20	`$i++`	4	**3**			
21	`while($i < 6)`	True				
22	`if ($i % 2 == 1)`	True				
23	`$s += 1`	**5**	3			
24	`$i++`	5	**4**			
25	`while($i < 6)`	True				
26	`if ($i % 2 == 1)`	False				
27	`$s += sss($i)`			**4**	?	?
28	`$sum = 0`			4	**0**	?
29	`$k = 1`			4	0	**1**
30	`$k <= $a`			True		
31	`$sum += $k`			4	**1**	1
32	`$k++`			4	1	**2**
33	`$k <= $a`			True		
34	`$sum += $k`			4	**3**	2
35	`$k++`			4	3	**3**
36	`$k <= $a`			True		
37	`$sum += $k`			4	**6**	4
38	`$k++`			4	6	**4**
39	`$k <= $a`			True		
40	`$sum += $k`			4	**10**	4
41	`$k++`			4	10	**5**
42	`$k <= $a`			False		
43	`return $sum`	**15**	4			
44	`$i++`	15	**5**			
45	`while($i < 6)`	True				
46	`if ($i % 2 == 1)`	True				
47	`$s += 1`	**16**	5			
48	`$i++`	16	**6**			
49	`while($i < 6)`	False				
50	`echo $s`	16 is displayed				

4. Solution

Step	Statement	Main Code				Function custom_div()	
		$k	$m	$a	$x	$b	$d
1	`$k = trim(fgets(STDIN))`	**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, $a)`					2	1
7	`return intval(($b + $d) / 2)`	12	2	1	**4**		
8	`echo $m, " ", $a, " ", $x`	2 1 4 is displayed					
9	`$a += 2`	12	2	**3**	4		
10	`$m++`	12	**3**	3	4		
11	`while ($a < 6)`	True					
12	`if ($k % $m != 0)`	False					
13	`$x = $a + $m + custom_div($m, $a)`					3	3
14	`return intval(($b + $d) / 2)`	12	3	3	**9**		
15	`echo $m, " ", $a, " ", $x`	3 3 9 is displayed					
16	`$a += 2`	12	3	**5**	9		
17	`$m++`	12	**4**	5	9		
18	`while ($a < 6)`	True					
19	`if ($k % $m != 0)`	False					
20	`$x = $a + $m + custom_div($m, $a)`					4	5
21	`return intval(($b + $d) / 2)`	12	4	5	**13**		
22	`echo $m, " ", $a, " ", $x`	4 5 13 is displayed					
23	`$a += 2`	12	4	**7**	13		
24	`$m++`	12	**5**	7	13		
25	`while ($a < 6)`	False					

5. Solution

```
function my_round($x) {
  $digit_to_check = intval(($x * 1000)) % 10;
  if ($digit_to_check >= 5) {
    $return_value = (intval(($x * 100)) + 1) / 100;
  }
  else {
    $return_value = (intval($x * 100)) / 100;
  }
  return $return_value;
}
```

6. Solution

```
<?php
  function find_min($a, $b) {
    $min = $a;
    if ($b < $min) {
      $min = $b;
    }
    return $min;
  }

  echo "Enter four numbers: ";
  $x1 = trim(fgets(STDIN));
  $x2 = trim(fgets(STDIN));
  $x3 = trim(fgets(STDIN));
  $x4 = trim(fgets(STDIN));

  //First approach
  $temp1 = find_min($x1, $x2);
  $temp2 = find_min($x3, $x4);
  echo find_min($temp1, $temp2);

  //Second approach
  echo find_min(find_min($x1, $x2), find_min($x3, $x4));
?>
```

7. Solution

```
<?php
  function Kelvin_to_Fahrenheit($kelvin) {
    return 1.8 * $kelvin - 459.67;
  }

  function Kelvin_to_Celsius($kelvin) {
    return $kelvin - 273.15;
  }

  echo "Enter a temperature in degrees Kelvin: ";
  $k = trim(fgets(STDIN));
  echo "Fahrenheit: ", Kelvin_to_Fahrenheit($k);
  echo "\n";
  echo "Celsius: ", Kelvin_to_Celsius($k);
?>
```

8. Solution

```
<?php
  function bmi($w, $h) {
    $b = $w * 703 / pow($h ,2);
    if ($b < 16) {
      $return_value = "You must add weight.";
    }
    elseif ($b < 18.5) {
      $return_value = "You should add some weight.";
    }
    elseif ($b < 25) {
      $return_value = "Maintain your weight.";
    }
    elseif ($b < 30) {
      $return_value = "You should lose some weight.";
    }
    else {
      $return_value = "You must lose weight.";
    }
    return $return_value;
  }

  echo "Enter your weight (in pounds): ";
  $weight = trim(fgets(STDIN));
  while (is_numeric($weight) != true || $weight < 0) {
    echo "Error! Enter your weight (in pounds): ";
    $weight = trim(fgets(STDIN));
  }

  echo "Enter your age: ";
  $age = trim(fgets(STDIN));
  while (is_numeric($age) != true || $age < 18) {
    echo "Error! Enter your age: ";
    $age = trim(fgets(STDIN));
  }

  echo "Enter your height (in inches): ";
  $height = trim(fgets(STDIN));
  while (is_numeric($height) != true || $height < 0) {
    echo "Error! Enter your height (in inches): ";
    $height = trim(fgets(STDIN));
  }
```

```
echo bmi($weight, $height);
?>
```

Chapter 40

40.5 Review Questions: True/False

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

40. 6 Review Exercises

1. Solution

Step	Statement	Main Code		Function `display()`
		$i	$x	$a
1	`$i = 1`	**1**	?	
2	`$i <= 5`	True		
3	`$x = trim(fgets(STDIN))`	1	**3**	
4	`display($x)`			**3**
5	`if ($a % 2 == 0)`			False
6	`echo $a, " is odd"`	The message “3 is odd” is displayed		
7	`$i++`	**2**	3	
8	`$i <= 5`	True		
9	`$x = trim(fgets(STDIN))`	2	**7**	
10	`display($x)`			**7**
11	`if ($a % 2 == 0)`			False
12	`echo $a, " is odd"`	The message “7 is odd” is displayed		
13	`$i++`	**3**	7	
14	`$i <= 5`	True		
15	`$x = trim(fgets(STDIN))`	3	**9**	
16	`display($x)`			**9**
17	`if ($a % 2 == 0)`			False
18	`echo $a, " is odd"`	The message “9 is odd” is displayed		
19	`$i++`	**4**	9	
20	`$i <= 5`	True		
21	`$x = trim(fgets(STDIN))`	4	**2**	

22	`display($x)`			**2**
23	`if ($a % 2 == 0)`			True
24	`echo $a, " is even"`	The message “2 is even” is displayed		
25	`$i++`	**5**	2	
26	`$i <= 5`	True		
27	`$x = trim(fgets(STDIN))`	5	**4**	
28	`display($x)`			**4**
29	`if ($a % 2 == 0)`			True
30	`echo $a, " is even"`	The message “4 is even” is displayed		
31	`$i++`	**6**	4	
32	`$i <= 5`	False		

2. *Solution*

Step	Statement	Main Code		Function `division()`	
		$x	$y	$a	$b
1	`$x = 20`	**20**	?		
2	`$y = 30`	20	**30**		
3	`while ($x % $y < 30)`	True			
4	`division($y, $x)`			**30**	**20**
5	`$b = intval($b / $a)`			30	**0**
6	`echo $a * $b, "\n"`	0 is displayed			
7	`$x = 4 * $y`	**120**	30		
8	`$y++`	120	**31**		
9	`while ($x % $y < 30)`	True			
10	`division($y, $x)`			**31**	**120**
11	`$b = intval($b / $a)`			31	**3**
12	`echo $a * $b, "\n"`	93 is displayed			
13	`$x = 4 * $y`	**124**	31		
14	`$y++`	124	**32**		
15	`while ($x % $y < 30)`	True			
16	`division($y, $x)`			**32**	**124**
17	`$b = intval($b / $a)`			32	**3**
18	`echo $a * $b, "\n"`	96 is displayed			
19	`$x = 4 * $y`	**128**	32		

20	`$y++`	128	**33**		
21	`while ($x % $y < 30)`	True			
22	`division($y, $x)`			**33**	**128**
23	`$b = intval($b / $a)`			33	**3**
24	`echo $a * $b, "\n"`	99 is displayed			
25	`$x = 4 * $y`	**132**	33		
26	`$y++`	132	**34**		
27	`while ($x % $y < 30)`	False			

3. Solution

Step	**Statement**	**Main Code**		**Function** `calculate()`		
		$i	**$m**	**$n**	**$s**	**$j**
1	`$i = 1`	**1**	?			
2	`$i <= 3`	True				
3	`$m = trim(fgets(STDIN));`	1	**2**			
4	`calculate($m)`			**2**	?	?
5	`$s = 0`			2	**0**	?
6	`$j = 2`			2	0	**2**
7	`$j <= 2 * n`			True		
8	`$s = $s + pow($j, 2)`			2	**4**	2
9	`$j += 2`			2	4	**4**
10	`$j <= 2 * n`			True		
11	`$s = $s + pow($j, 2)`			2	**20**	4
12	`$j += 2`			2	20	**6**
13	`$j <= 2 * n`			False		
14	`echo $s`	20 is displayed				
15	`$i++`	**2**	2			
16	`$i <= 3`	True				
17	`$m = trim(fgets(STDIN));`	2	**3**			
18	`calculate($m)`			3	?	?
19	`$s = 0`			3	**0**	?
20	`$j = 2`			3	0	**2**
21	`$j <= 2 * n`			True		
22	`$s = $s + pow($j, 2)`			3	**4**	2
23	`$j += 2`			3	4	**4**

24	`$j <= 2 * n`			True		
25	`$s = $s + pow($j, 2)`			3	**20**	4
26	`$j += 2`			3	20	**6**
27	`$j <= 2 * n`			True		
28	`$s = $s + pow($j, 2)`			3	**56**	6
29	`$j += 2`			3	56	**8**
30	`$j <= 2 * n`			False		
31	`echo $s`	56 is displayed				
32	`$i++`	**3**	3			
33	`$i <= 3`	True				
34	`$m = trim(fgets(STDIN));`	3	**4**			
35	`calculate($m)`			**4**	?	?
36	`$s = 0`			4	**0**	?
37	`$j = 2`			4	0	**2**
38	`$j <= 2 * n`			True		
39	`$s = $s + pow($j, 2)`			4	**4**	2
40	`$j += 2`			4	4	**4**
41	`$j <= 2 * n`			True		
42	`$s = $s + pow($j, 2)`			4	**20**	4
43	`$j += 2`			4	20	**6**
44	`$j <= 2 * n`			True		
45	`$s = $s + pow($j, 2)`			4	**56**	6
46	`$j += 2`			4	56	**8**
47	`$j <= 2 * n`			True		
48	`$s = $s + pow($j, 2)`			4	**120**	8
49	`$j += 2`			4	120	**10**
50	`$j <= 2 * n`			False		
51	`echo $s`	120 is displayed				
52	`$i++`	**4**	4			
53	`$i <= 3`	False				

4. Solution

```
function maximum($a, $b, $c, $d, $e) {
  $max = $a;
  if ($b > $max) {
    $max = $b;
  }
  if ($c > $max) {
    $max = $c;
  }
  if ($d > $max) {
    $max = $d;
  }
  if ($e > $max) {
    $max = $e;
  }
  echo $max;
}
```

5. Solution

```
<?php
  function num_of_days($year, $month) {
    switch ($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;
    }

    echo $days, "\n";
  }

  echo "Enter a year: ";
  $y = trim(fgets(STDIN));
  for ($m = 1; $m <= 12; $m++) {
    num_of_days($y, $m);
  }
?>
```

6. Solution

```
<?php
  function display_menu() {
    echo "\n";
    echo "1. Convert meters to miles\n";
    echo "2. Convert miles to meters\n";
    echo "3. Exit\n";
    echo "Enter a choice: ";
  }

  function meters_to_miles($meters) {
    echo $meters, " meters equals ", $meters / 1609.344 ," miles";
  }

  function miles_to_meters($miles) {
    echo $miles, " miles equals ", $miles * 1609.344 ," meters";
  }

  do {
    display_menu();

    $choice = trim(fgets(STDIN));

    if ($choice == 3) {
      echo "Bye!";
    }
    else {
      echo "Enter distance: ";
      $distance = trim(fgets(STDIN));
      if ($choice == 1) {
        meters_to_miles($distance);
      }
      else {
        miles_to_meters($distance);
      }
    }
  } while ($choice != 3);
?>
```

7. Solution

```
<?php
  function amount_to_pay($seconds) {
    if ($seconds <= 600) {
      $extra = 0;
    }
    elseif ($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;

    echo "Total amount to pay: ", $total;
  }

  echo "Enter number of seconds: ";
  $seconds = trim(fgets(STDIN));
  amount_to_pay($seconds);
?>
```

Chapter 41

41.9 Review Questions: True/False

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

41.10 Review Exercises

1. Solution

The value 5 is displayed

2. Solution

The value 14 is displayed

3. Solution

The value 14 is displayed

4. Solution

Step	Statement	Main Code				Function swap()		
		a	m	k	x	x	y	temp
1	`$k = trim(fgets(STDIN))`	?	?	**12**	?			
2	`$m = 1`	?	**1**	12	?			
3	`$a = 1`	**1**	1	12	?			
4	`while ($a < 8)`	True						
5	`if ($k % $m != 0)`	False						
6	`$x = $a + $m + intval($a - $m)`	1	1	12	**2**			
7	`echo $m, " ", $a, " ", $x, "\n"`	1 1 2 is displayed						
8	`$a += 2`	**3**	1	12	2			
9	`$m++`	3	**2**	12	2			

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

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

5. Solution

“hellohellohello” is displayed

6. Solution

The value 15 is displayed

7. Solution

11 4 is displayed

8. Solution

```
<?php
  define("STUDENTS", 10);
  define("LESSONS", 5);

  function part1(&$names, &$grades) {
    for ($i = 0; $i <= STUDENTS - 1; $i++) {
      echo "Enter name for student No. ", $i + 1, ": ";
      $names[$i] = trim(fgets(STDIN));
      for ($j = 0; $j <= LESSONS - 1; $j++) {
        echo "Enter grade for lesson No. ", $j + 1, ": ";
        $grades[$i][$j] = trim(fgets(STDIN));
      }
    }
  }

  function part2($grades) {
    $average = array();
    for ($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;
  }

  function part3(&$average, &$names) {
    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 = $names[$n];
          $names[$n] = $names[$n - 1];
          $names[$n - 1] = $temp;
        }
        elseif ($average[$n] == $average[$n - 1]) {
          if ($names[$n] < $names[$n - 1]) {
            $temp = $names[$n];
            $names[$n] = $names[$n - 1];
            $names[$n - 1] = $temp;
          }
        }
      }
    }
  }

  $names = array();
  $grades = array(array());
  part1($names, $grades);
```

```
  $average = part2($grades);

  part3($average, $names);

  for ($i = 0; $i <= STUDENTS - 1; $i++) {
    echo $names[$i], "\t", $average[$i], "\n";
  }
?>
```

9. Solution

```
<?php
  function part1() {
    echo "Enter a message: ";
    $message = trim(fgets(STDIN));
    $message = strtolower($message);
    return $message;
  }

  function part2($message) {
    $last_pos = strlen($message) - 1;
    return $last_pos;
  }

  function part3($message, $last_pos) {
    $message_clean = "";
    for ($i = 0; $i <= $last_pos; $i++) {
      if ($message[$i] != " " && $message[$i] != "," &&
          $message[$i] != "." && $message[$i] != "?") {

        $message_clean .= $message[$i];
      }
    }
    return $message_clean;
  }

  function part4($message_clean) {
    $middle_pos = intval((strlen($message_clean) - 1) / 2);
    $j = strlen($message_clean) - 1;
    $palindrome = true;
    for ($i = 0; $i <= $middle_pos; $i++) {
      if ($message_clean[$i] != $message_clean[$j]) {
        $palindrome = false;
        break;
      }
      $j--;
    }
    return $palindrome;
  }

  function part5($message) {
    $last_pos = part2($message);
    $message_clean = part3($message, $last_pos);
    $palindrome = part4($message_clean);
    return $palindrome;
  }

  $message = part1();
  $palindrome = part5($message);
  if ($palindrome == true) {
    echo "The message is palindrome";
  }
?>
```

10. Solution

```
<?php
  $a = trim(fgets(STDIN));
  $b = trim(fgets(STDIN));
  $c = trim(fgets(STDIN));

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

  echo $max;
?>
```

11. Solution

```
function f1($a, $b, $c, &$sum, &$average) {
  $sum = $a + $b + $c;
  $average = $sum / 3;
}
```

12. Solution

```
function my_round($x, $decimal_places = 2) {
  $digit_to_check = intval(($x * pow(10, $decimal_places + 1))) % 10;
  if ($digit_to_check >= 5) {
    $return_value = (intval(($x * pow(10, $decimal_places))) + 1) / pow(10, $decimal_places);
  }
  else {
    $return_value = (intval($x * pow(10, $decimal_places))) / pow(10, $decimal_places);
  }
  return $return_value;
}
```

13. Solution

```
<?php
  function get_input() {
    do {
      echo "Enter Yes or No: ";
      $answer = strtoupper(trim(fgets(STDIN)));
    } while ($answer != "YES" && $answer != "NO");
    return $answer;
  }

  function find_area($base, $height) {
    return $base * $height;
  }

  do {
    echo "Enter the base of the parallelogram: ";
    $base = trim(fgets(STDIN));
    echo "Enter the height of the parallelogram: ";
    $height = trim(fgets(STDIN));

    echo "Area = ", find_area($base, $height), "\n";

    echo "Would you like to repeat? ";
  } while (get_input() == "YES") ;
?>
```

14. Solution

```
<?php
  define("STUDENTS", 100);

  function get_arrays(&$names, &$grades) {
    for ($i = 0; $i <= STUDENTS - 1; $i++) {
      echo "Enter name: ";
      $names[$i] = trim(fgets(STDIN));
      echo "Enter grade: ";
      $grades[$i] = trim(fgets(STDIN));
    }
  }

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

  function sort_arrays(&$grades, &$names) {
    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;
    }
  }

  $names = array();
  $grades = array();
  get_arrays($names, $grades);
  $average = get_average($grades);
  sort_arrays($grades, $names);
  for ($i = 0; $i <= STUDENTS - 1; $i++) {
    if ($grades[$i] < $average) {
      echo $names[$i];
    }
  }
?>
```

15. Solution

```
<?php
  define("JUDGES", 10);

  function get_array() {
    $score = array();
    for ($i = 0; $i <= JUDGES - 1; $i++) {
      echo "Judge No ", $i + 1 ,". Enter score: ";
      $score[$i] = trim(fgets(STDIN));
    }
    return $score;
  }

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

  echo "Enter artist's name: ";
  $name = trim(fgets(STDIN));
  $score = get_array();
  find_min_max($score, $min, $max);

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

  $points = $sum - $min - $max;
  echo "Artist ", $name ," got ", $points ," points";
?>
```

16. Solution

```
<?php
  function woc($index) {
    if ($index == 1) {
      $return_value = 1;
    }
    else {
      $return_value = 2 * woc($index - 1);
    }
    return $return_value;
  }
  $sum = 0;
  for ($i = 1; $i <= 64; $i++) {
    $sum += woc($i);
  }
  echo $sum;
?>
```

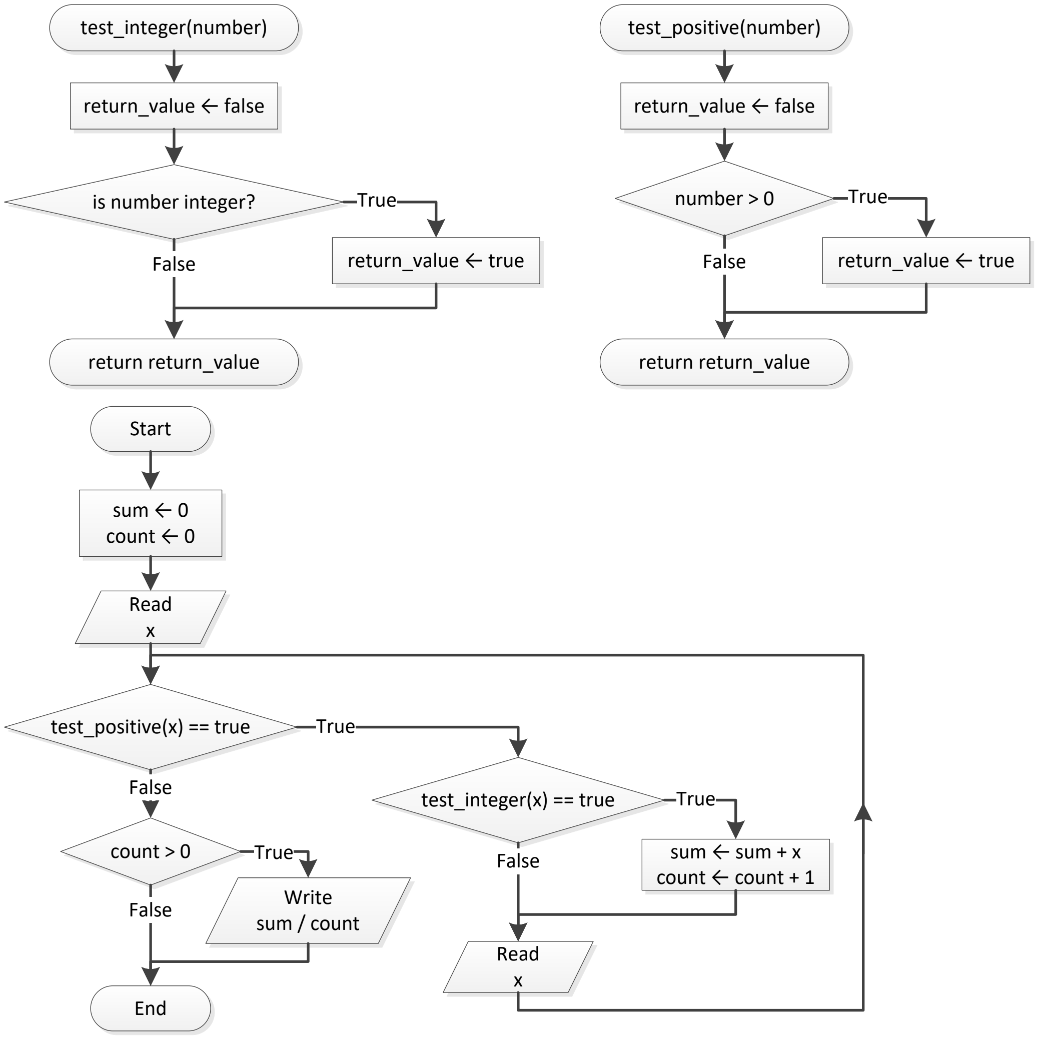
17. Solution

```
<?php
  function fact($value) {
    if ($value == 1) {
      $return_value = 1;
    }
    else {
      $return_value = $value * fact($value - 1);
    }

    return $return_value;
  }

  function my_cos($x, $i = 40) {
    if ($i == 0) {
      $return_value = 1;
    }
    else {
      $return_value = my_cos($x, $i - 4) + pow($x, $i) / fact($i) - pow($x, $i - 2) / fact($i - 2);
    }

    return $return_value;
  }

  echo my_cos(pi()/4);
?>
```

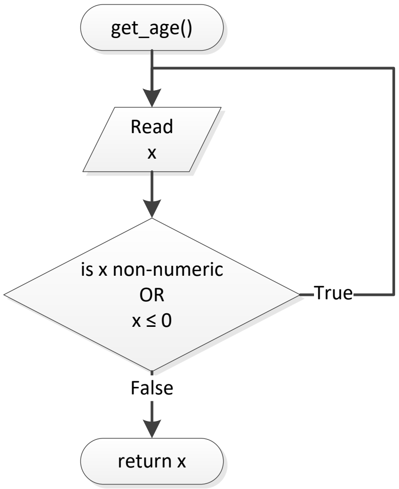
Chapter 42

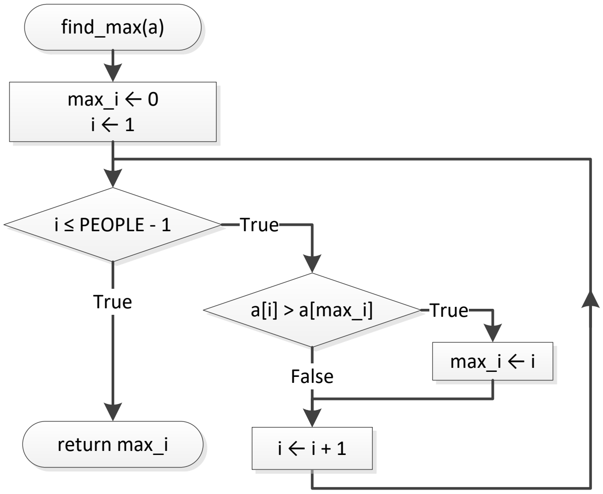
42.4 Review Exercises

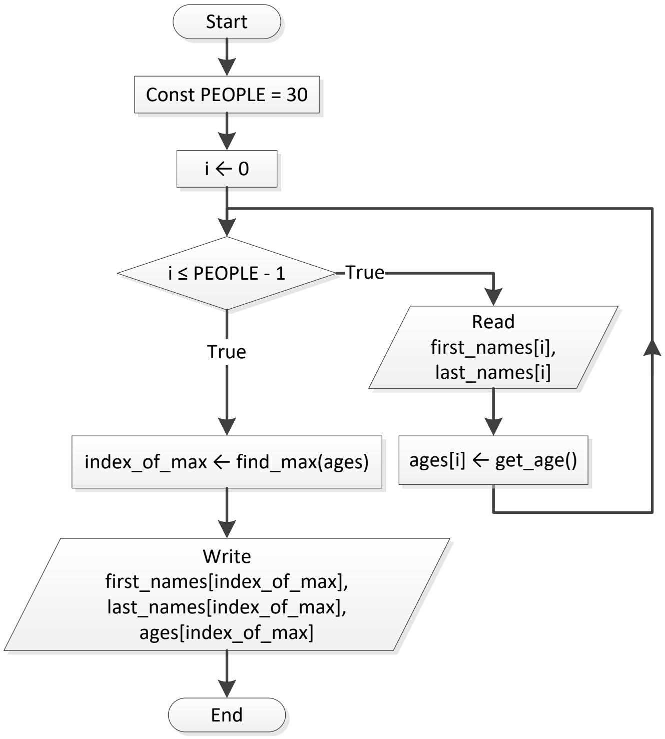
1. *Solution*



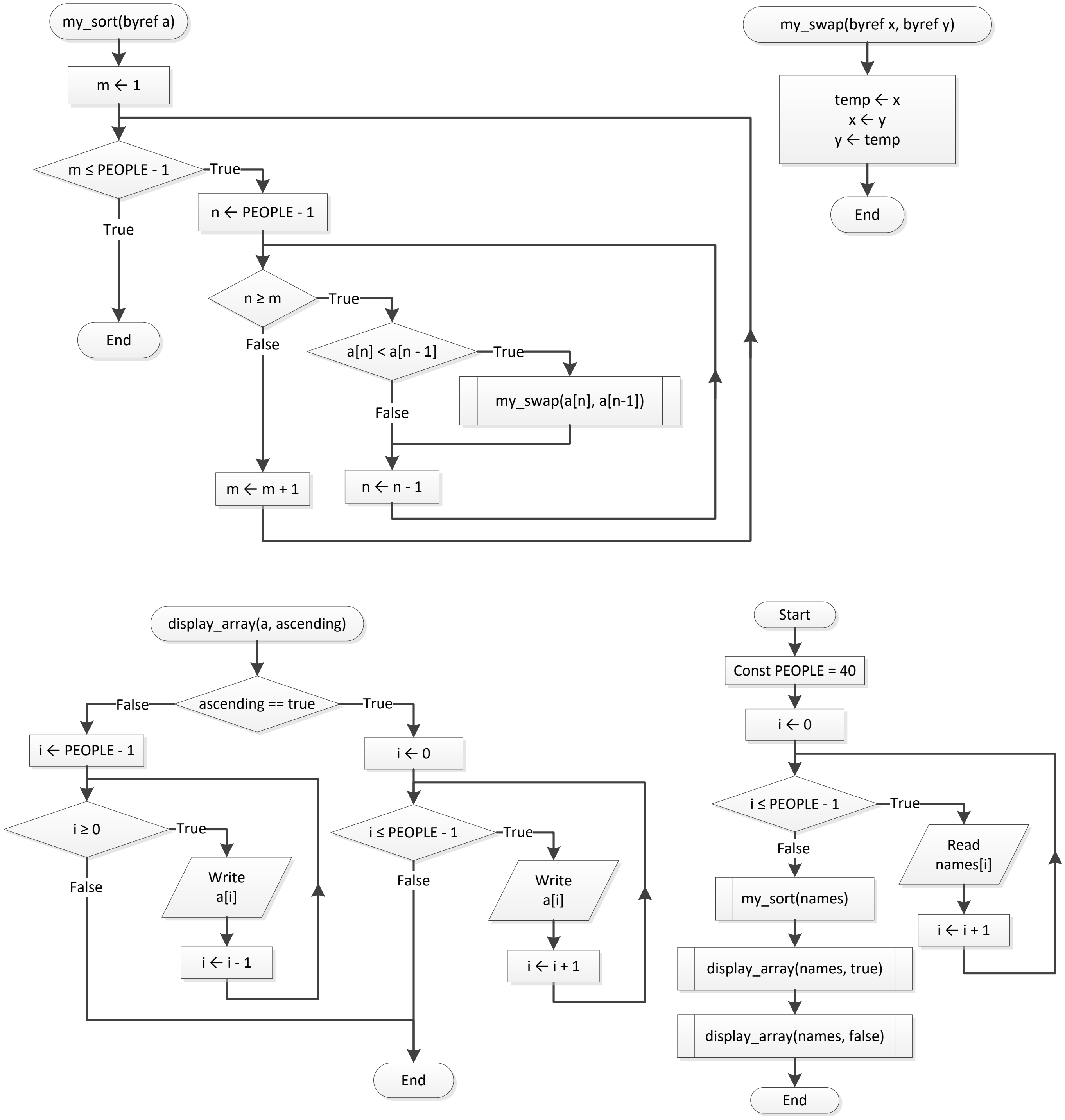
2. *Solution*


get_age()
Read
x
is x non-numeric
OR
x ≤ 0
True
False
return x

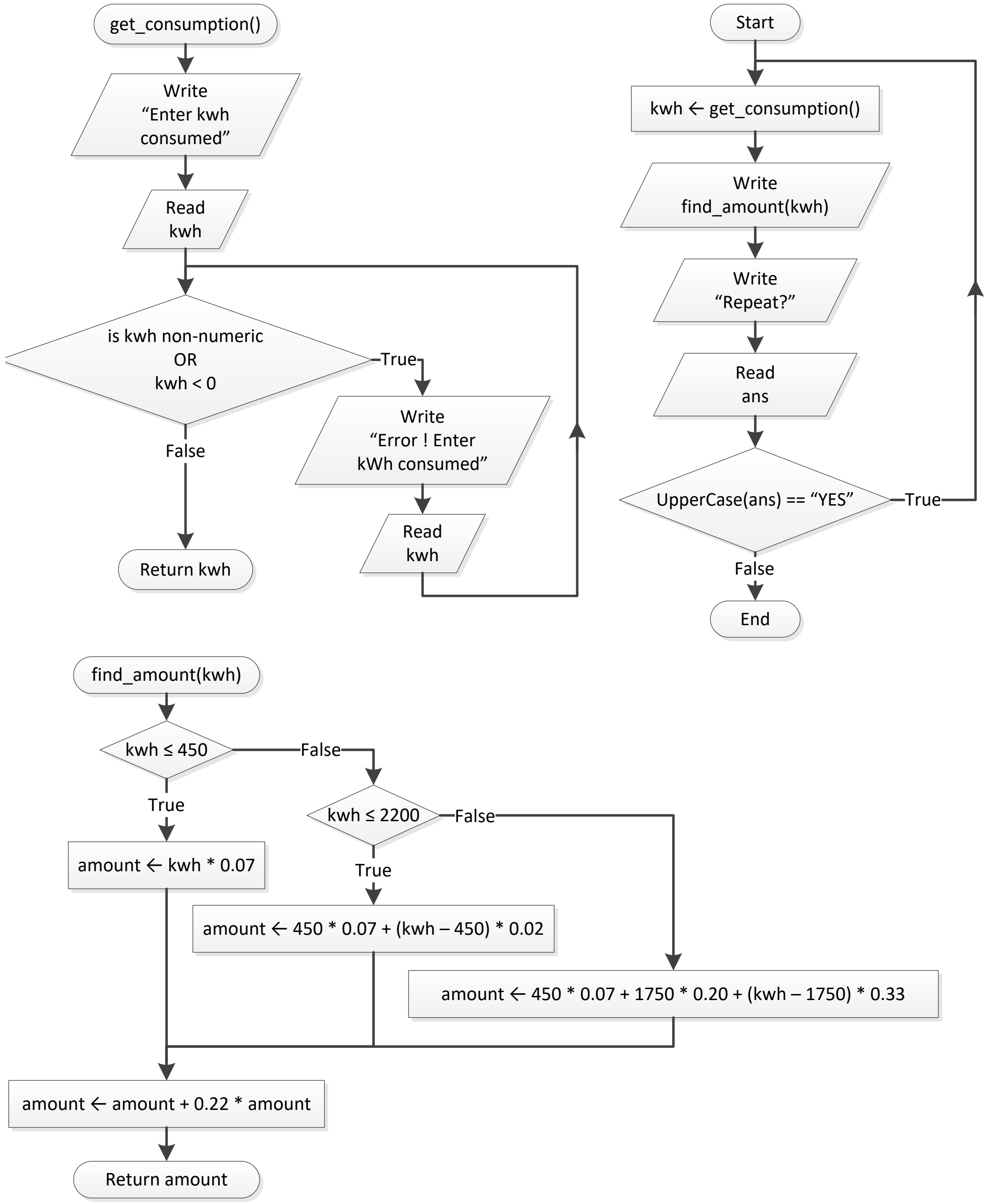

find_max(a)
max_i ← 0
i ← 1
i ≤ PEOPLE - 1
True
True
a[i] > a[max_i]
True
max_i ← i
False
return max_i
i ← i + 1


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

3. Solution


my_sort(byref a)
m ← 1
m ≤ PEOPLE - 1
True
n ← PEOPLE - 1
True
End
n ≥ m
True
False
a[n] < a[n - 1]
True
my_swap(a[n], a[n-1])
False
m ← m + 1
n ← n - 1
my_swap(byref x, byref y)
temp ← x
x ← y
y ← temp
End
display_array(a, ascending)
ascending == true
False
True
i ← PEOPLE - 1
i ← 0
i ≥ 0
True
Write
a[i]
False
i ← i - 1
i ≤ PEOPLE - 1
True
Write
a[i]
False
i ← i + 1
End
Start
Const PEOPLE = 40
i ← 0
i ≤ PEOPLE - 1
True
Read
names[i]
False
my_sort(names)
i ← i + 1
display_array(names, true)
display_array(names, false)
End

4. Solution



5. Solution

```
<?php
  define("STUDENTS", 20);
  define("LESSONS", 10);

  function get_arrays(&$names, &$grades) {
    for ($i = 0; $i <= STUDENTS - 1; $i++) {
      $names[$i] = trim(fgets(STDIN));
      for ($j = 0; $j <= LESSONS - 1; $j++) {
        $grades[$i][$j] = trim(fgets(STDIN));
      }
    }
  }

  function find_average($grades) {
    for ($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;
  }

  function display($names, $average) {
    for ($i = 0; $i <= STUDENTS - 1; $i++) {
      if ($average[$i] > 89) {
        echo $names[$i], $average[$i];
      }
    }
  }

  $names = array();
  $grades = array(array());
  $av = array();

  get_arrays($names, $grades);
  $av = find_average($grades);
  display($names, $av);
?>
```

6. Solution

```
<?php
  function fib($n) {
    if ($n == 0 || $n == 1) {
      $return_val = $n;
    }
    else {
      $return_val = fib($n - 1) + fib($n - 2);
    }

    return $return_val;
  }

  do {
    $n = trim(fgets(STDIN));
    while ($n < 0) {
      echo "Error";
      $n = trim(fgets(STDIN));
    }

    echo fib($n);
    echo "Again? ";
    $ans = trim(fgets(STDIN));
  } while ($ans == "Y");
?>
```

Chapter 43

43.3 Review Exercises

1. *Solution*

```
<?php
  define("ACCURACY", 0.000000001);

  function factorial($n) {
    $return_value = 1;
    for ($i = 1; $i <= $n; $i++) {
      $return_value *= $i;
    }
    return $return_value;
  }

  function my_sin($x) {
    $sign = 1;
    $sinus = 0;
    $i = 1;
    do {
      $sinus_previous = $sinus;
      $sinus += $sign * pow($x, $i) / factorial($i);

      $sign = -$sign;
      $i += 2;
    } while (abs($sinus - $sinus_previous) > ACCURACY);
    return $sinus;
  }

  function degrees_to_rad($degrees) {
    return 2 * pi() * $degrees / 360;
  }

  for ($i = 0; $i <= 360; $i++) {
    echo "sin(", $i ,") ~= ", my_sin(degrees_to_rad($i)), "\n";
  }
?>
```

2. *Solution*

```
<?php
  function is_leap($year) {
    $return_value = false;
    if ($year % 4 == 0 && $year % 100 != 0 || $year % 400 == 0) {
```

```
    $return_value = true;
  }
  return $return_value;
}

function num_of_days($year, $month) {
  switch ($month) {
    case 4:
    case 6:
    case 9:
    case 11:
      $days = 30;
      break;
    case 2:
      if (is_leap($year) == true) {
        $days = 29;
      }
      else {
        $days = 28;
      }
      break;
    default:
      $days = 31;
  }

  return $days;
}

function check_date($day, $month, $year) {
  $return_value = true;
  if (is_numeric($month) != true || $month < 1 || $month > 12) {
    $return_value = false;
  }
  elseif (is_numeric($year) != true) {
    $return_value = false;
  }
  elseif (is_numeric($day) != true || $day < 1 || $day > num_of_days($year, $month)) {
    $return_value = false;
  }
  return $return_value;
}

echo "Enter day: ";
$day = trim(fgets(STDIN));
echo "Enter month: ";
$month = trim(fgets(STDIN));
```

```
  echo "Enter year: ";
  $year = trim(fgets(STDIN));
  while (check_date($day, $month, $year) == false) {
    echo "Error!\n";
    echo "Enter day: ";
    $day = trim(fgets(STDIN));
    echo "Enter month: ";
    $month = trim(fgets(STDIN));
    echo "Enter year: ";
    $year = trim(fgets(STDIN));
  }

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

  echo $sum;
?>
```

3. Solution

```
<?php
  function dice() {
    return rand(1, 6);
  }

  echo "Player1 - Enter name: ";
  $name1 = trim(fgets(STDIN));
  echo "Player2 - Enter name: ";
  $name2 = trim(fgets(STDIN));

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

      $dice1 =  dice();
      $dice2 =  dice();
      echo $dice1, "   ", $dice2, "\n";
      $sum += $dice1 + $dice2;
    }
    if ($player == 1) {
      $sum_player1 = $sum;
    }
    else {
      $sum_player2 = $sum;
    }
  }

  if ($sum_player1 == $sum_player2) {
    echo "Tie!";
  }
  elseif ($sum_player1 > $sum_player2) {
    echo $name1 . " wins";
  }
  else {
    echo $name2 . " wins";
  }
?>
```

4. Solution

```
<?php
  define("GAS", 1);
  define("DIESEL", 2);
  define("HYBRID", 3);
  define("TAX_RATE", 0.10);
  define("CARS", 40);

  function get_choice() {
    echo "1. Gas\n";
    echo "2. Diesel\n";
    echo "3. Hybrid\n";
    echo "Enter type of the car: ";
    $type = trim(fgets(STDIN));
    return $type;
  }

  function get_days() {
    echo "Enter total number of rental days: ";
    $rental_days = trim(fgets(STDIN));
    return $rental_days;
  }

  function get_charge($type, $rental_days) {
```

```
  if ($type == GAS) {
    if ($rental_days <= 5) {
      $charge = $rental_days * 24;
    }
    elseif ($rental_days <= 8) {
      $charge = 5 * 24 + ($rental_days - 5) * 22;
    }
    else {
      $charge = 5 * 24 + 3 * 22 + ($rental_days - 8) * 18;
    }
  }
  elseif ($type == DIESEL) {
    if ($rental_days <= 5) {
      $charge = $rental_days * 28;
    }
    elseif ($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 = $rental_days * 30;
    }
    elseif ($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;
}

$rented_car_types = array();
$rented_days = array();

for ($i = 0; $i <= CARS - 1; $i++) {
   $rented_car_types[$i] = get_choice();
   $rented_days[$i] = get_days();
}

$sum = 0;
```

```
  for ($i = 0; $i <= CARS - 1; $i++) {
     $charge = get_charge($rented_car_types[$i], $rented_days[$i]);
     echo "Car No ", $i + 1, ": ", $charge, "\n";
     $sum += $charge;
  }

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

  echo "Hybrids rented: ", $count ,"\n";
  echo "Net profit: ", $sum / (1 + TAX_RATE);
?>
```

5. Solution

```
<?php
  define("CHANNELS", 10);
  define("DAYS", 7);

  $day_names = array("Monday", "Tuesday", "Wednesday", "Thursday", "Friday", "Saturday", "Sunday");

  function get_data(&$names, &$viewers) {
    global $day_names;

    for ($i = 0; $i <= CHANNELS - 1; $i++) {
      echo "Enter name for channel No. ", $i + 1, ": ";
      $names[$i] = trim(fgets(STDIN));
      for ($j = 0; $j <= DAYS - 1; $j++) {
        echo "Enter the number of viewers of the main news program on ", $day_names[$j];
        echo " for channel ", $names[$i], ": ";
        $viewers[$i][$j] = trim(fgets(STDIN));
      }
    }
  }

  function get_average($a) {
    $sum = 0;
    for ($i = 0; $i <= 4; $i++) {
      $sum += $a[$i];
    }
    return $sum / 5;
  }
  $names = array();
  $viewers = array(array());
  get_data($names, $viewers);

  for ($i = 0; $i <= CHANNELS - 1; $i++) {
    $temporary_array = array();
    for ($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)) {
      echo $names[$i], "\n";
    }
  }

  for ($i = 0; $i <= CHANNELS - 1; $i++) {
    $increasing = true;
    for ($j = 1; $j <= DAYS - 1; $j++) {
      if ($viewers[$i][$j] <= $viewers[$i][$j - 1]) {
        $increasing = false;
      }
    }
    if ($increasing == true) {
      echo $names[$i], "\n";
    }
  }
?>
```

6. *Solution*

```
<?php
  define("CITIZENS", 30);

  function input_data(&$SSNs, &$answers) {
    for ($i = 0; $i <= CITIZENS - 1; $i++) {
      echo "Enter SSN: ";
      $SSNs[$i] = trim(fgets(STDIN));
      echo "Enter answer: ";
      $answers[$i] = trim(fgets(STDIN));
    }
  }

  function sort_arrays(&$SSNs, &$answers) {
    for ($m = 0; $m <= CITIZENS - 1; $m++) {
      $min = $SSNs[$m];
      $index_of_min = $m;
      for ($n = $m; $n <= CITIZENS - 1; $n++) {
```

```
      if ($SSNs[$n] < $min) {
        $min = $SSNs[$n];
        $index_of_min = $n;
      }
    }
    $temp = $SSNs[$m];
    $SSNs[$m] = $SSNs[$index_of_min];
    $SSNs[$index_of_min] = $temp;
    $temp = $answers[$m];
    $answers[$m] = $answers[$index_of_min];
    $answers[$index_of_min] = $temp;
  }
}

function search_array($SSNs, $SSN) {
  $left = 0;
  $right = CITIZENS - 1;
  $found = false;
  while ($left <= $right && $found == false) {
    $middle = intval(($left + $right) / 2);

    if ($SSNs[$middle] > $SSN) {
      $right = $middle - 1;
    }
    elseif ($SSNs[$middle] < $SSN) {
      $left = $middle + 1;
    }
    else {
      $found = true;
      $index_position = $middle;
    }
  }

  if ($found == false) {
    echo "SSN not found!";
    $return_value = -1;
  }
  else {
    $return_value = $index_position;
  }
  return $return_value;
}

function count_answers($answers, $answer) {
  $count = 0;
  for ($i = 0; $i <= CITIZENS - 1; $i++) {
      if ($answers[$i] == $answer) {
        $count++;
      }
    }
    return $count;
  }

  do {
    $SSNs = array();
    $answers = array();
    input_data($SSNs, $answers);
    sort_arrays($SSNs, $answers);

    echo "Enter an SSN to search: ";
    $SSN = trim(fgets(STDIN));

    $index = search_array($SSNs, $SSN);
    if ($index != -1) {
      $answer = $answers[$index];
      echo $answer, "\n";

      $count = count_answers($answers, $answer);
      echo $count * 100 / CITIZENS;
    }
    echo "Repeat? ";
    $answer = trim(fgets(STDIN));
  } while ($answer == "Yes");

?>
```

7. Solution

```
<?php
  define("TEAMS", 8);
  define("GAMES", 12);

  function input_data(&$names, &$results)  {
    for ($i = 0; $i <= TEAMS - 1; $i++) {
      echo "Enter team name: ";
      $names[$i] = trim(fgets(STDIN));
      for ($j = 0; $j <= GAMES - 1; $j++) {
        echo "Enter result (W, L, T): ";
        $results[$i][$j] = trim(fgets(STDIN));
      }
    }
  }
```

```
function display_result($names, $results) {
  echo "Enter a result to search (W, L, T): ";
  $result = trim(fgets(STDIN));
  for ($i = 0; $i <= TEAMS - 1; $i++) {
    echo "Team: ", $names[$i], "\n";
    $found = false;
    for ($j = 0; $j <= GAMES - 1; $j++) {
      if ($results[$i][$j] == $result) {
        echo "Week: ", $j + 1, "\n";
        $found = true;
      }
    }
    if ($found == false) {
      echo "Nothing found\n";
    }
  }
}

function find_team($names) {
  echo "Enter a name to search: ";
  $name = trim(fgets(STDIN));

  $i = 0;
  while ($i < TEAMS - 1 && $names[$i] != $name) {
    $i++;
  }

  if ($names[$i] != $name) {
    $return_value = -1;
  }
  else {
    $return_value = $i;
  }
  return $return_value;
}


$names = array();
$results = array(array());
input_data($names, $results);
display_result($names, $results);

$index = find_team($names);
while ($index != -1) {
  $sum = 0;
  for ($j = 0; $j <= GAMES - 1; $j++) {
      if ($results[$index][$j] == "W") {
        $sum += 3;
      }
      elseif ($results[$index][$j] == "T") {
        $sum += 1;
      }
    }
    echo "Points: ", $sum, "\n";
    $index = find_team($names);
  }
?>
```

8. Solution

```
<?php
  //space is a valid character!
  $alphabet = " abcdefghijklmnopqrstuvwxyz";

  function my_encrypt($message, $encryption_key) {
    global $alphabet;

    $return_value = "";
    for ($i = 0; $i <= strlen($message) - 1; $i++) {
      $index = strpos($alphabet, $message[$i]);
      $new_index = ($index + $encryption_key) % 27; //26 letters + 1 space
      $return_value .= $alphabet[$new_index];
    }
    return $return_value;
  }

  function my_decrypt($message, $decryption_key) {
    global $alphabet;

    $return_value = "";
    for ($i = 0; $i <= strlen($message) - 1; $i++) {
      $index = strpos($alphabet, $message[$i]);
      $new_index = ($index + 27 - $decryption_key) % 27; //26 letters + 1 space
      $return_value .= $alphabet[$new_index];
    }
    return $return_value;
  }

  function display_menu() {
    echo "\n";
    echo "1. Encrypt a message\n";
    echo "2. Decrypt a message\n";
    echo "3. Exit\n";
  }

  do {
    display_menu();
    echo "Enter a choice: ";
    $choice = trim(fgets(STDIN));
    if ($choice == 1) {
      echo "Enter a message to encrypt: ";
      $message = trim(fgets(STDIN));
      echo "Enter an encryption key: ";
      $encryption_key = trim(fgets(STDIN));
      echo "Your encrypted message is: ", my_encrypt($message, $encryption_key);
    }
    elseif ($choice == 2) {
      echo "Enter a message to decrypt: ";
      $message = trim(fgets(STDIN));
      echo "Enter an decryption key: ";
      $decryption_key = trim(fgets(STDIN));
      echo "Your decrypted message is: ", my_decrypt($message, $decryption_key);
    }
  } while ($choice != 3);
?>
```

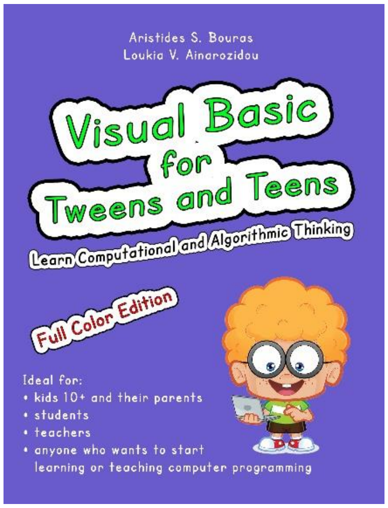
Some Final Words from the Authors

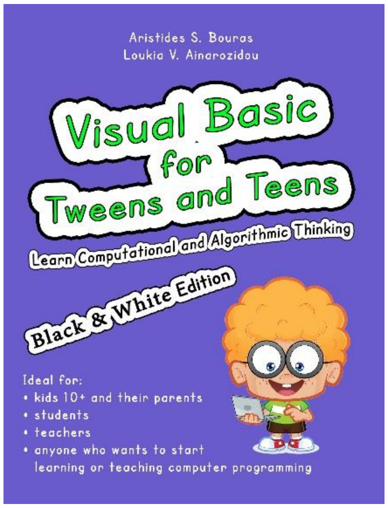
We hope you really enjoyed reading this book. We made every possible effort to make it comprehensible even by people that probably have no previous experience in programming.

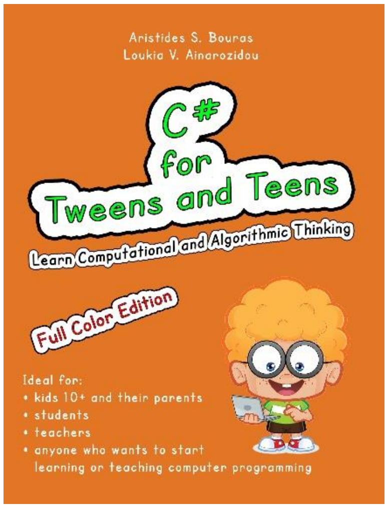
So if you liked this book, please visit the web store where you bought it and show us your gratitude by writing a good review and giving us as many stars as possible. By doing this, you will encourage us to continue writing and of course you'll help other readers to reach us.

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

More of our Books


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

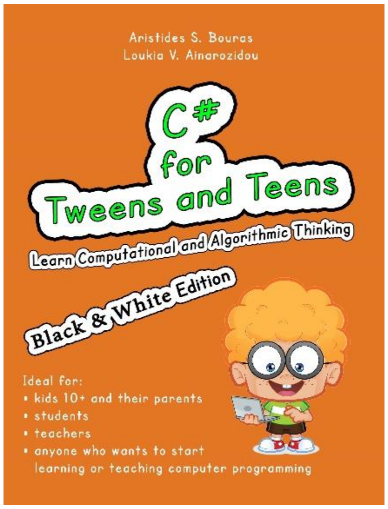

Aristides S. Bouras
Loukia V. Ainarozidou
Visual Basic for Tweens and Teens
Learn Computational and Algorithmic Thinking
Black & White Edition
Ideal for:
• kids 10+ and their parents
• students
• teachers
• anyone who wants to start learning or teaching computer programming

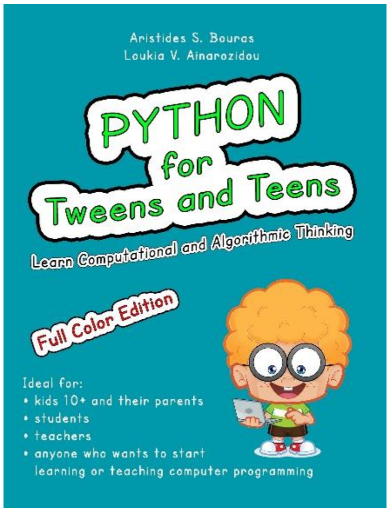

Aristides S. Bouras
Loukia V. Ainarozidou
C# for Tweens and Teens
Learn Computational and Algorithmic Thinking
Full Color Edition
Ideal for:
• kids 10+ and their parents
• students
• teachers
• anyone who wants to start learning or teaching computer programming

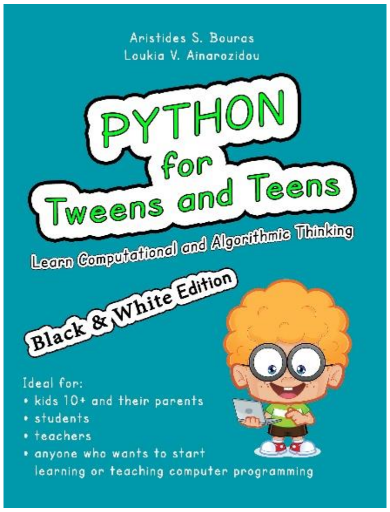
ISBN-10: 1982083670
ISBN-13: 978-1982083670

ISBN-10: 1982083697
ISBN-13: 978-1982083694

ISBN-10: 1973727684
ISBN-13: 978-1973727682


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

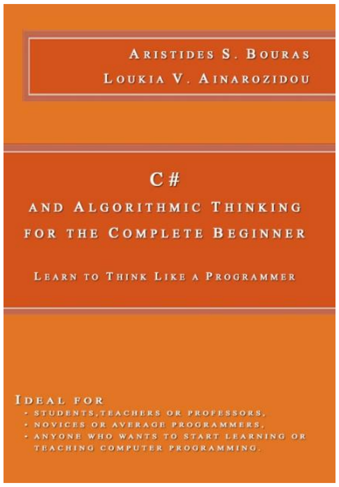

Aristides S. Bouras
Loukia V. Ainarozidou
PYTHON for Tweens and Teens
Learn Computational and Algorithmic Thinking
Full Color Edition
Ideal for:
• kids 10+ and their parents
• students
• teachers
• anyone who wants to start learning or teaching computer programming


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

ISBN-10: 1973727765
ISBN-13: 978-1973727767

ISBN-10: 1543127940
ISBN-13: 978-1543127942

ISBN-10: 1546611215
ISBN-13: 978-1546611219



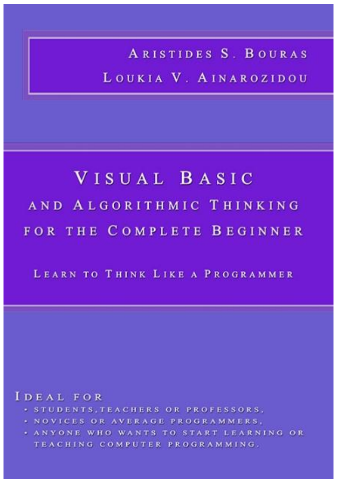
Aristides S. Bouras
Loukia V. Ainarozidou

C#
and Algorithmic Thinking
for the Complete Beginner

Learn to Think Like a Programmer

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

ISBN-10: 1508952485
ISBN-13: 978-1508952480



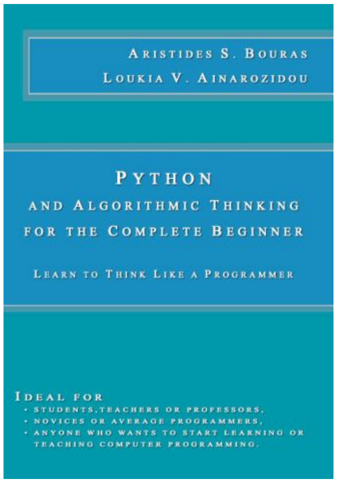
Aristides S. Bouras
Loukia V. Ainarozidou

Visual Basic
and Algorithmic Thinking
for the Complete Beginner

Learn to Think Like a Programmer

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

ISBN-10: 1511798963
ISBN-13: 978-1511798969



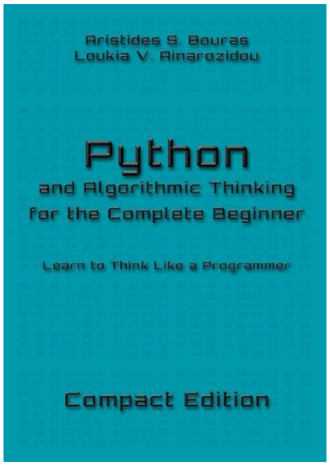
Aristides S. Bouras
Loukia V. Ainarozidou

Python
and Algorithmic Thinking
for the Complete Beginner

Learn to Think Like a Programmer

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

ISBN-10: 1514802163
ISBN-13: 978-1514802168



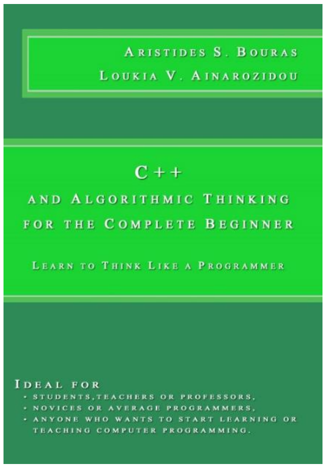
Aristides S. Bouras
Loukia V. Ainarozidou

Python
and Algorithmic Thinking
for the Complete Beginner

Learn to Think Like a Programmer

Compact Edition

ISBN-10: 1983276111
ISBN-13: 978-1983276118



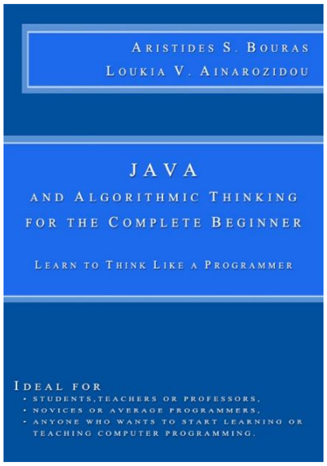
Aristides S. Bouras
Loukia V. Ainarozidou

C++
and Algorithmic Thinking
for the Complete Beginner

Learn to Think Like a Programmer

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

ISBN-10: 1508577552
ISBN-13: 978-1508577553



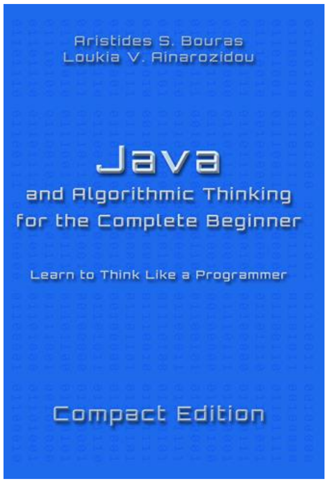
Aristides S. Bouras
Loukia V. Ainarozidou

Java
and Algorithmic Thinking
for the Complete Beginner

Learn to Think Like a Programmer

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

ISBN-10: 1506179398
ISBN-13: 978-1506179391



Aristides S. Bouras
Loukia V. Ainarozidou

Java
and Algorithmic Thinking
for the Complete Beginner

Learn to Think Like a Programmer

Compact Edition

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