

PHP
AND ALGORITHMIC THINKING
FOR THE COMPLETE BEGINNER
Second Edition

The Answers

Aristides S. Bouras

Copyright ©

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

RCode: 221008

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

Warning and Disclaimer

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

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

Table of Contents

How to Report Errata.....	7
.....	8
1.7 Review Questions: True/False.....	8
1.8 Review Questions: Multiple Choice.....	8
Review in “Introductory Knowledge”	9
Chapter 1 Review Crossword Puzzles	9
.....	11
4.16 Review Questions: True/False	11
Chapter 4 4.17 Review Questions: Multiple Choice	11
.....	12
5.8 Review Questions: True/False.....	12
Chapter 5 5.9 Review Questions: Multiple Choice.....	12
5.10 Review Exercises	12
.....	13
Chapter 6 6.4 Review Questions: True/False.....	13
6.5 Review Questions: Multiple Choice.....	13
Chapter 7	14
7.7 Review Questions: True/False.....	14
7.8 Review Questions: Multiple Choice.....	14
Chapter 8 7.9 Review Exercises	14
.....	15
Chapter 9 8.2 Review Questions: True/False.....	15
8.3 Review Exercises	15
.....	17
Chapter 10 9.4 Review Exercises	17
Chapter Review in “Getting Started with PHP”	19
Review Crossword Puzzles	19
.....	20
10.2 Review Exercises	20
.....	25
11.3 Review Questions: True/False	25
11.4 Review Questions: Multiple Choice.....	25
11.5 Review Exercises	25

	27
12.2 Review Exercises		27
	29
13.2 Review Exercises		29
	32
Chapter 12 14.4 Review Questions: True/False		32
Chapter 14 14.5 Review Questions: Multiple Choice		32
Chapter 14 14.6 Review Exercises		33
Review in “Sequence Control Structures”		35
Review Crossword Puzzle		35
	36
15.9 Review Questions: True/False		36
Chapter 15 15.10 Review Questions: Multiple Choice		36
15.11 Review Exercises		36
	38
Chapter 16 16.2 Review Questions: True/False		38
16.3 Review Questions: Multiple Choice		38
16.4 Review Exercises		38
Chapter 17		44
17.2 Review Questions: True/False		44
17.3 Review Questions: Multiple Choice		44
Chapter 18 17.4 Review Exercises		44
	51
Chapter 19 18.2 Review Questions: True/False		51
18.3 Review Exercises		51
Chapter 20		60
19.2 Review Questions: True/False		60
Chapter 21 19.3 Review Exercises		60
	67
Chapter 22 20.3 Review Questions: True/False		67
20.4 Review Exercises		67
	72
21.4 Review Exercises		72
	78
22.9 Review Questions: True/False		78

22.10 Review Questions: Multiple Choice	78
22.11 Review Exercises.....	79
.....	83
23.6 Review Exercises	83
Review in “Decision Control Structures”	102
Review Crossword Puzzle	102
Chapter 23	103
24.3 Review Questions: True/False	103
.....	104
Chapter 24 24.4 Review Questions: True/False	104
Chapter 25 25.5 Review Questions: Multiple Choice.....	104
Chapter 25 25.6 Review Exercises	104
.....	115
Chapter 26 26.3 Review Questions: True/False	115
26.4 Review Questions: Multiple Choice.....	115
26.5 Review Exercises	115
Chapter 27	129
27.3 Review Questions: True/False	129
27.4 Review Questions: Multiple Choice.....	129
Chapter 28 27.5 Review Exercises	129
.....	137
28.8 Review Questions: True/False	137
Chapter 29 28.9 Review Questions: Multiple Choice.....	137
Chapter 30 28.10 Review Exercises.....	137
.....	142
29.4 Review Exercises	142
.....	151
Chapter 31 30.7 Review Questions: True/False	151
30.8 Review Exercises	151
Review in “Loop Control Structures”	172
Review Crossword Puzzle	172
.....	173
31.13 Review Questions: True/False	173
31.14 Review Questions: Multiple Choice	173
31.15 Review Exercises.....	173

.....	187
32.7 Review Questions: True/False	187
32.8 Review Questions: Multiple Choice	187
32.9 Review Exercises	188
Chapter 32	200
33.7 Review Questions: True/False	200
33.8 Review Questions: Multiple Choice	200
33.9 Review Exercises	200
Chapter 33	210
34.7 Review Questions: True/False	210
34.8 Review Exercises	211
Chapter 34	211
Review in “Arrays in PHP”	247
Review Crossword Puzzle	247
.....	248
35.4 Review Questions: True/False	248
Chapter 35	249
Chapter 36	249
36.8 Review Questions: True/False	249
36.9 Review Exercises	249
Chapter 37	261
37.9 Review Questions: True/False	261
Chapter 37	261
37.10 Review Exercises	261
.....	270
38.3 Review Exercises	270
Chapter 38	270
Chapter 39	285
Review in “Subprograms”	285
Review Crossword Puzzle	285
.....	286
39.9 Review Questions: True/False	286
39.10 Review Exercises	286
Review in “Object Oriented Programming”	298
Review Crossword Puzzle	298
Some Final Words from the Author	299

How to Report Errata

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

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

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

Chapter 1

1.7 Review Questions: True/False

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

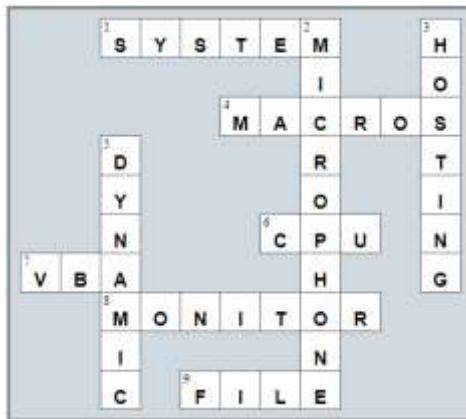
1.8 Review Questions: Multiple Choice

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

Review Crossword Puzzles

[illegible][illegible]

3.



Chapter 4

4.16 Review Questions: True/False

- | | |
|-----------|-----------|
| 1. true | 22. true |
| 2. false | 23. false |
| 3. false | 24. false |
| 4. false | 25. false |
| 5. false | 26. false |
| 6. true | 27. true |
| 7. false | 28. true |
| 8. true | 29. false |
| 9. true | 30. false |
| 10. true | 31. false |
| 11. false | 32. false |
| 12. false | 33. false |
| 13. true | 34. true |
| 14. true | 35. false |
| 15. false | 36. false |
| 16. true | 37. false |
| 17. false | 38. false |
| 18. false | 39. true |
| 19. false | 40. true |
| 20. true | 41. false |
| 21. true | |

4.17 Review Questions: Multiple Choice

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

Chapter 5

5.8 Review Questions: True/False

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

5.9 Review Questions: Multiple Choice

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

5.10 Review Exercises

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

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

6.4 Review Questions: True/False

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

Chapter 6

6.5 Review Questions: Multiple Choice

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

Chapter 7

7.7 Review Questions: True/False

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

7.8 Review Questions: Multiple Choice

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

7.9 Review Exercises

- ii, iv, v, ix, x
- i. String, ii. Boolean, iii. String, iv. String, v. Float, vi. Integer
- i. d, ii. f, iii. c, iv. e
- i. 26, ii. 28
- i. 5, ii. 6
- i. 1, ii. 0, iii. 1, iv. 1, v. 0, vi. 1
- i. $2 * 3$, ii. 4
- i. 2, ii. 0, iii. 1, iv. 0, v. 0, vi. 0
- i. 2, ii. 10
- My name is George Malkovich
- i. (-3) , ii. 1
- California California

8.2 Review Questions: True/False

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

Chapter 8 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	It displays: 40, 1, 1, 40			

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	It displays: 49, 10, 3, 1470			

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	It displays: 28, 2, 2, 112			

2. Solution

For the input values of 8, 4

Step	Statement	\$a	\$b	\$c	\$d	\$e
1	\$a = trim(fgets(STDIN))	8	?	?	?	?
2	\$b = trim(fgets(STDIN))	8	4	?	?	?

3	<code>\$c = \$a + \$b</code>	8	4	12	?	?
4	<code>\$d = 1 + \$a / \$b * \$c + 2</code>	8	4	12	27	?
5	<code>\$e = \$c + \$d</code>	8	4	12	27	39
6	<code>\$c += \$d + \$e</code>	8	4	78	27	39
7	<code>\$e--</code>	8	4	78	27	38
8	<code>\$d -= \$c + \$d % \$c</code>	8	4	78	-78	38
9	<code>echo \$c, ", ", \$d, ", ", \$e</code>	It displays: 78, -78, 38				

For the input values of 4, 4

Step	Statement	\$a	\$b	\$c	\$d	\$e
1	<code>\$a = trim(fgets(STDIN))</code>	4	?	?	?	?
2	<code>\$b = trim(fgets(STDIN))</code>	4	4	?	?	?
3	<code>\$c = \$a + \$b</code>	4	4	8	?	?
4	<code>\$d = 1 + \$a / \$b * \$c + 2</code>	4	4	8	11	?
5	<code>\$e = \$c + \$d</code>	4	4	8	11	19
6	<code>\$c += \$d + \$e</code>	4	4	38	11	19
7	<code>\$e--</code>	4	4	38	11	18
8	<code>\$d -= \$c + \$d % \$c</code>	4	4	38	-38	18
9	<code>echo \$c, ", ", \$d, ", ", \$e</code>	It displays: 38, -38, 18				

9.4 Review Exercises

1. Solution

Chapter 9

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

2. Solution

For the input values of 5, 5

Step	Statement	\$a	\$b	\$c	\$d	\$e
1	<code>\$a = trim(fgets(STDIN))</code>	5	?	?	?	?
2	<code>\$b = trim(fgets(STDIN))</code>	5	5	?	?	?
3	<code>\$c = \$a + \$b</code>	5	5	10	?	?
4	<code>\$d = 5 + \$a / \$b * \$c + 2</code>	5	5	10	17	?
5	<code>\$e = \$c - \$d</code>	5	5	10	17	-7
6	<code>\$c += \$d + \$c</code>	5	5	37	17	-7
7	<code>\$e--</code>	5	5	37	17	-8
8	<code>\$d += \$e + \$c % \$b</code>	5	5	37	11	-8
9	<code>echo \$c, ", ", \$d, ", ", \$e</code>	It displays: 37, 11, -8				

For the input values of 4, 2

Step	Statement	\$a	\$b	\$c	\$d	\$e
1	<code>\$a = trim(fgets(STDIN))</code>	4	?	?	?	?
2	<code>\$b = trim(fgets(STDIN))</code>	4	2	?	?	?
3	<code>\$c = \$a + \$b</code>	4	2	6	?	?
4	<code>\$d = 5 + \$a / \$b * \$c + 2</code>	4	2	6	19	?
5	<code>\$e = \$c - \$d</code>	4	2	6	19	-13
6	<code>\$c += \$d + \$c</code>	4	2	31	19	-13
7	<code>\$e--</code>	4	2	31	19	-14
8	<code>\$d += \$e + \$c % \$b</code>	4	2	31	6	-14
9	<code>echo \$c, ", ", \$d, ", ", \$e</code>	It displays: 31, 6, -14				

3. Solution

For the input value of 5

Step	Statement	\$a	\$b	\$c
1	<code>\$b = trim(fgets(STDIN))</code>	?	5	?
2	<code>\$c = 5</code>	?	5	5
3	<code>\$c = \$c * \$b</code>	?	5	25
4	<code>\$a = 3 * \$c % 10</code>	5	5	25

5	echo \$a	It displays: 5
----------	----------	----------------

For the input value of 4

Step	Statement	\$a	\$b	\$c
1	\$b = trim(fgets(STDIN))	?	4	?
2	\$c = 5	?	4	5
3	\$c = \$c * \$b	?	4	20
4	\$a = 3 * \$c % 10	0	4	20
5	echo \$a	It displays: 0		

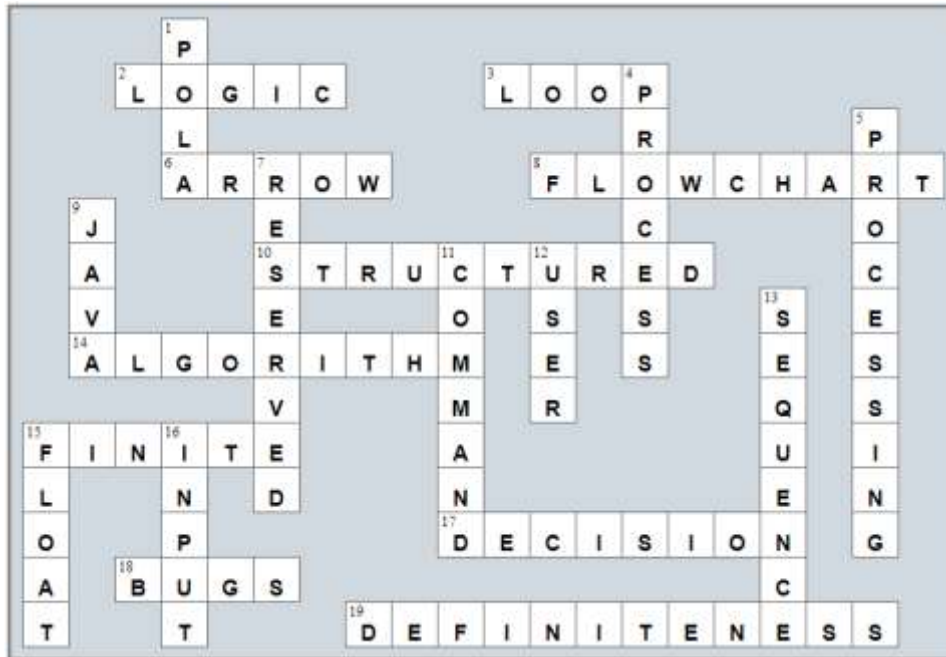
For the input value of 15

Step	Statement	\$a	\$b	\$c
1	\$b = trim(fgets(STDIN))	?	15	?
2	\$c = 5	?	15	5
3	\$c = \$c * \$b	?	15	75
4	\$a = 3 * \$c % 10	5	15	75
5	echo \$a	It displays: 5		

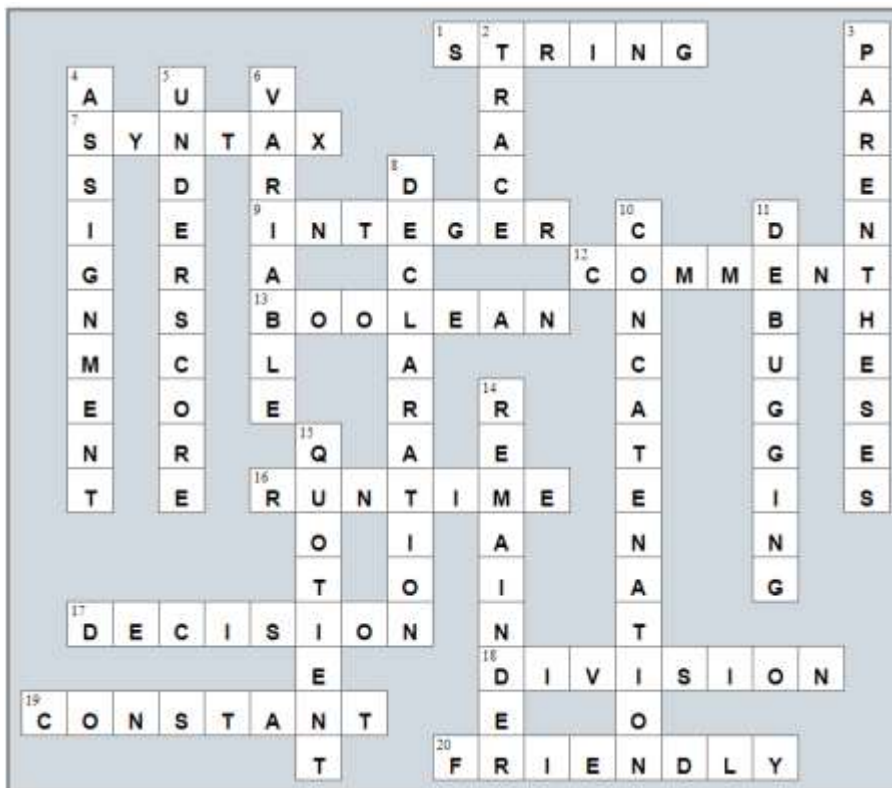
Review in "Getting Started with PHP"

Review Crossword Puzzles

1.



2.



10.2 Review Exercises

1. Solution

Chapter 10

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

    $area = 0.5 * $b * $h;

    echo $area, "\n";
?>
```

2. Solution

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

    $angle3 = 180 - $angle1 - $angle2;

    echo $angle3, "\n";
?>
```

3. Solution

```
<?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) / 4.0;

    echo $average, "\n";
?>
```

4. Solution

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

```
$perimeter = 2 * pi() * $r;  
  
echo $perimeter, "\n";  
?>
```

5. Solution

```
<?php  
echo "Enter diameter (in meters): ";  
$d = trim(fgets(STDIN));  
  
$radius = $d / 2;  
$volume = 4 / 3 * pi() * $radius ** 3;  
  
echo $volume, "\n";  
?>
```

6. Solution

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

7. Solution

```
<?php  
echo "Enter diameter: ";  
$d = trim(fgets(STDIN));  
  
$radius = $d / 2;  
$perimeter = 2 * pi() * $radius;  
$area = PI * $radius ** 2;  
$volume = 4 / 3 * pi() * $radius ** 3;  
  
echo $radius, " ", $perimeter, " ", $area, " ", $volume, "\n";  
?>
```

8. 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, "\n";  
?>
```

9. 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 ** 2;

echo $s, "\n";
?>
```

10. Solution

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

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

echo $c, "\n";
?>
```

11. Solution

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

$bmi = $w * 703.0 / $h ** 2;

echo $bmi, "\n";
?>
```

12. Solution

```
<?php
echo "Enter subtotal: ";
$s_total = trim(fgets(STDIN));
echo "Enter gratuity rate (0 - 100): ";
$g_rate = trim(fgets(STDIN));

$tip = $s_total * $g_rate / 100;

$total = $s_total + $tip;

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

13. 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;

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

echo $avg, "\n";
?>

```

14. Solution

```

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

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

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

echo $btax_price, "\n";
?>

```

15. Solution

```

<?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, "\n";
?>

```

16. Solution

```

<?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, "\n";  
?>
```

17. Solution

```
<?php  
echo "Enter current month: ";  
$month = trim(fgets(STDIN));  
echo "Enter current day: ";  
$day = trim(fgets(STDIN));  
  
$days_passed = ($month - 1) * 30 + $day;  
$days_left = 360 - $days_passed;  
  
echo $days_left, "\n";  
?>
```


Chapter 11

11.3 Review Questions: True/False

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

11.4 Review Questions: Multiple Choice

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

11.5 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 + 20.4	33.4	?	?
3	\$b = round(\$a) % 4	33.4	1	?
4	\$c = \$b % 3	33.4	1	1
5	echo \$a, ", ", \$b, ", ", \$c	It displays: 33.4, 1, 1		

For the input value of 4

Step	Statement	\$a	\$b	\$c
1	\$a = trim(fgets(STDIN))	4	?	?
2	\$a += 6 / sqrt(\$a) * 2 + 20.4	30.4	?	?
3	\$b = round(\$a) % 4	30.4	2	?
4	\$c = \$b % 3	30.4	2	2
5	echo \$a, ", ", \$b, ", ", \$c	It displays: 30.4, 2, 2		

2. Solution

For the input value of -2

Step	Statement	\$a	\$b	\$c
1	\$a = trim(fgets(STDIN))	-2	?	?
2	\$b = abs(\$a) % 4 + (int)(\$a ** 4)	-2	18	?
3	\$c = \$b % 5	-2	18	3
4	echo \$b, ", ", \$c	It displays: 18, 3		

For the input value of -3

Step	Statement	\$a	\$b	\$c
1	<code>\$a = trim(fgets(STDIN))</code>	-3	?	?
2	<code>\$b = abs(\$a) % 4 + (int)(\$a ** 4)</code>	-3	84	?
3	<code>\$c = \$b % 5</code>	-3	84	4
4	<code>echo \$b, ", ", \$c</code>	It displays: 84, 4		

3. Solution

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

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

    echo $degrees, "\n";
?>
```

4. Solution

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

    $hypotenuse = sqrt($a ** 2 + $b ** 2);

    echo $hypotenuse, "\n";
?>
```

5. Solution

```
<?php
    echo "Enter angle  $\theta$  (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, "\n";
?>
```

12.2 Review Exercises

1. Solution

Chapter 12

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

2. Solution

- i. $y = (x + 3) ** (5 * w) / (7 * (x - 4))$
- ii. $y = (3 * x ** 2 - x ** 3 / 4) ** (1 / 5)$
- iii. $y = \text{sqrt}(x ** 4 - 2 * x ** 3 - 7 * x ** 2 + x) / (4 * (7 * x ** 4 - 3 / 4 * x ** 3) * (7 * x ** 2 + x)) ** (1 / 3)$
- iv. $y = x / (x - 3 * (x - 1)) + x * (x - 1) ** (1 / 5) / ((x ** 3 - 2) * (x - 1) ** 3)$
- v. $y = (\sin(\pi() / 3) - \cos(\pi() / 2 * w)) ** 2$
- vi. $y = (\sin(\pi() / 2 * x) + \cos(3 * \pi() / 2 * w)) ** 3 / (\tan(2 * \pi() / 3 * w) - \sin(\pi() / 2 * x)) ** 0.5 + 6$

3. Solution

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

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

echo $y, "\n";
?>
```

4. Solution

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

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

echo $y, "\n";
?>
```

5. Solution

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

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

```
    echo $y, "\n";
?>
```

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 * (3 * $w + 1) ** (1 / 5) * (5 * $x + 4) / (($x ** 3 +
3) * ($x - 1) ** 7);

    echo $y, "\n";
?>
```

7. Solution

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

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

    echo $y, "\n";
?>
```

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, "\n";
?>
```

13.2 Review Exercises

1. Solution

Chapter 13

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

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

echo $result, "\n";
?>
```

2. Solution

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

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

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

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

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

$reversed_number = $digit5 * 10000 + $digit4 * 1000 + $digit3 * 100 + $digit2 * 10 + $digit1;
echo $reversed_number, "\n";
?>
```

3. Solution

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

$result = $n % 2;

echo $result, "\n";
?>
```

4. Solution

```
<?php
echo "Enter an integer: ";
```

```

    $n = trim(fgets(STDIN));

    $result = 1 - $n % 2;

    echo $result, "\n";
?>

```

5. Solution

```

<?php
    echo "Enter an elapsed time in seconds: ";
    $number = trim(fgets(STDIN));

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

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

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

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

    echo $weeks, " week(s) ", $days, " day(s) ", $hours, " hour(s) ";
    echo $minutes, " minute(s) and ", $seconds, " second(s)\n";
?>

```

6. Solution

```

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

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

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

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

    echo $usd20, " note(s) of $20 ", $usd10, " note(s) of $10 ";
    echo $usd5, " note(s) of $5 and ", $usd1, " note(s) of $1\n";
?>

```

7. Solution

```

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

    $distance = $steps * 25;

```

```
$miles = (int)($distance / 63360);  
$r = $distance % 63360;  
  
$yards = (int)($r / 36);  
$r = $r % 36;  
  
$feet = (int)($r / 12);  
$inches = $r % 12;  
  
echo $miles, " mile(s) ", $yards, " yard(s) ";  
echo $feet, " foot/feet and ", $inches, " inch(es)\n";  
?>
```

Chapter 14

14.4 Review Questions: True/False

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

14.5 Review Questions: Multiple Choice

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

14.6 Review Exercises

1. Solution

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

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

    echo $random_word, "\n";
?>
```

3. Solution

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

    $x = str_replace(" ", "", strtolower($name));

    $secret_password = $x[rand(0, strlen($x) - 1)] .
                      $x[rand(0, strlen($x) - 1)] .
                      $x[rand(0, strlen($x) - 1)] .
                      rand(1000, 9999);

    echo $secret_password, "\n";
?>
```

4. Solution

First approach

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

    $s_number = (string)$number;

    $digit1 = $s_number[0];
    $digit2 = $s_number[1];
    $digit3 = $s_number[2];

    $reversed_number = 100 * (int)$digit3 + 10 * (int)$digit2 + (int)$digit1;

    echo $reversed_number, "\n";
?>
```

Second approach

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

    $s_number = (string)$number;
    $reversed_number = (int)($s_number[2] . $s_number[1] . $s_number[0]);

    echo $reversed_number, "\n";
?>
```

Review in "Sequence Control Structures"

Review Crossword Puzzle

1.



Chapter 15

15.9 Review Questions: True/False

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

15.10 Review Questions: Multiple Choice

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

15.11 Review Exercises

1. Solution

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

2. Solution

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

3. Solution

Boolean Expression1 (BE1)	Boolean Expression2 (BE2)	BE1 BE2	BE1 && BE2	!(BE2)
false	false	false	false	true
false	true	true	false	false
true	false	true	false	true
true	true	true	true	false

4. Solution

\$a	\$b	\$c	\$a > 3 \$c > \$b && \$c > 1	\$a > 3 && \$c > \$b \$c > 1
4	-6	2	true	true

-3	2	-4	false	false
2	5	5	false	true

5. Solution

Expression	Value
<code>(\$x + \$y) ** 3</code>	8
<code>(\$x + \$y) / (\$x ** 2 - 14)</code>	1
<code>\$x - 1 == \$y + 5</code>	true
<code>\$x > 2 && \$y == 1</code>	false
<code>\$x == 1 \$y == -2 && !(\$flag == false)</code>	true
<code>!(\$x >= 3) && (\$x % 2 > 1)</code>	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)`

16.2 Review Questions: True/False

1. false

2. false

3. true

4. false
5. false

6. false

7. true

8. false

16.3 Review Questions: Multiple Choice

1. b

2. c

3. d
4. d

5. d

16.4 Review Exercises

1. Solution

The corrections/additions are in red

```
<?php
$x = trim(fgets(STDIN));
$y = -5;
if ($x * $y / 2 > 20) {
    $y *= 2;
    $x += 4 * $x ** 2;
}

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

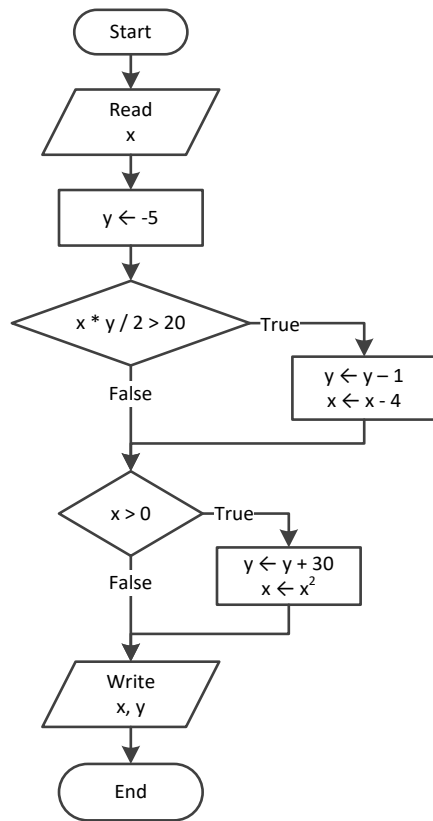
2. Solution

For the input value of 10

Step	Statement	\$x	\$y
1	<code>\$x = trim(fgets(STDIN))</code>	10	?
2	<code>\$y = -5</code>	10	-5
3	<code>if (\$x * \$y / 2 > 20)</code>	false	
4	<code>if (\$x > 0)</code>	true	
5	<code>\$y += 30</code>	10	25
6	<code>\$x = \$x ** 2</code>	100	25
7	<code>echo \$x, ", ", \$y</code>	It displays: 100, 25	

For the input value of -10

Step	Statement	\$x	\$y
1	<code>\$x = trim(fgets(STDIN))</code>	-10	?
2	<code>\$y = -5</code>	-10	-5
3	<code>if (\$x * \$y / 2 > 20)</code>	true	
4	<code>\$y--</code>	-10	-6
5	<code>\$x -= 4</code>	-14	-6
6	<code>if (\$x > 0)</code>	false	
7	<code>echo \$x, ", ", \$y</code>	It displays: -14, -6	



3. Solution

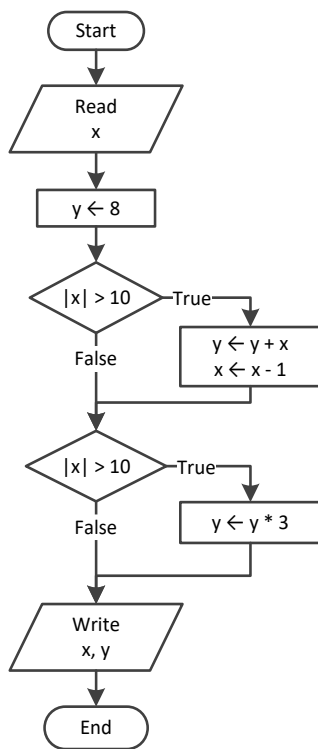
For the input value of -11

Step	Statement	\$x	\$y
1	<code>\$x = trim(fgets(STDIN))</code>	-11	?
2	<code>\$y = 8</code>	-11	8
3	<code>if (abs(\$x) > 10)</code>	true	
4	<code>\$y += \$x</code>	-11	-3
5	<code>\$x--</code>	-12	-3
6	<code>if (abs(\$x) > 10)</code>	true	
7	<code>\$y *= 3</code>	-12	-9

8	echo \$x, ", ", \$y	It displays: -12, -9
----------	---------------------	----------------------

For the input value of 11

Step	Statement	\$x	\$y
1	<code>\$x = trim(fgets(STDIN))</code>	11	?
2	<code>\$y = 8</code>	11	8
3	<code>if (abs(\$x) > 10)</code>	true	
4	<code>\$y += \$x</code>	11	19
5	<code>\$x--</code>	10	19
6	<code>if (abs(\$x) > 10)</code>	false	
7	echo \$x, ", ", \$y	It displays: 10, 19	



4. Solution

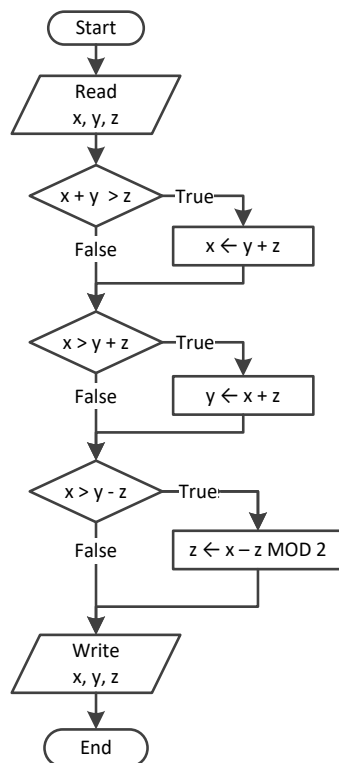
For input values of 1, 2 and 3

Step	Statement	\$x	\$y	\$z
1	<code>\$x = trim(fgets(STDIN))</code>	1	?	?
2	<code>\$y = trim(fgets(STDIN))</code>	1	2	?
3	<code>\$z = trim(fgets(STDIN))</code>	1	2	3
4	<code>if (\$x + \$y > \$z)</code>	false		
5	<code>if (\$x > \$y + \$z)</code>	false		
6	<code>if (\$x > \$y - \$z)</code>	true		
7	<code>\$z = \$x - \$z % 2</code>	1	2	0

8	<code>echo \$x, " ", \$y, " ", \$z</code>	It displays: 1, 2, 0
----------	---	----------------------

For input values of 4, 2 and 1

Step	Statement	\$x	\$y	\$z
1	<code>\$x = trim(fgets(STDIN))</code>	4	?	?
2	<code>\$y = trim(fgets(STDIN))</code>	4	2	?
3	<code>\$z = trim(fgets(STDIN))</code>	4	2	1
4	<code>if (\$x + \$y > \$z)</code>	true		
5	<code>\$x = \$y + \$z</code>	3	2	1
6	<code>if (\$x > \$y + \$z)</code>	false		
7	<code>if (\$x > \$y - \$z)</code>	true		
8	<code>\$z = \$x - \$z % 2</code>	3	2	2
9	<code>echo \$x, " ", \$y, " ", \$z</code>	It displays: 3, 2, 2		



5. Solution

```

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

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

```

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\n";
    }
?>
```

7. Solution

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

    if ($x > 14) {
        echo "You can drive a car in Kansas (USA)\n";
    }
?>
```

8. Solution

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

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

9. Solution

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

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

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!\n";
}
?>
```

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, "\n";
?>
```

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\n";
}
?>
```

17.2 Review Questions: True/False

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

Chapter 17

17.3 Review Questions: Multiple Choice

1. b
2. c
3. c

17.4 Review Exercises

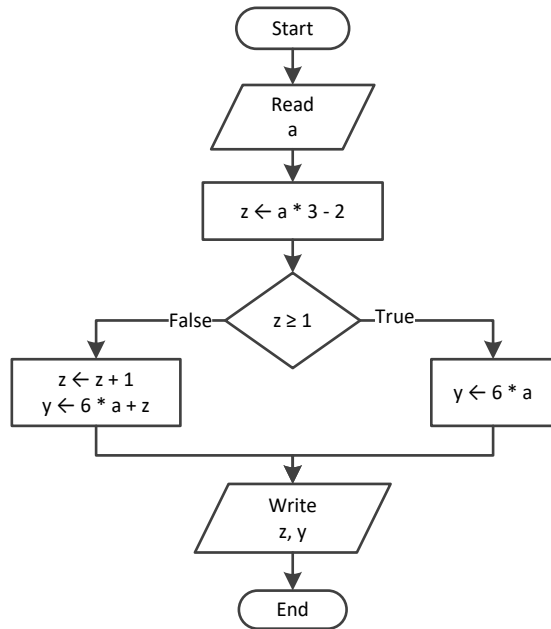
1. Solution

For input value of 3

Step	Statement	\$a	\$y	\$z
1	<code>\$a = trim(fgets(STDIN))</code>	3	?	?
2	<code>\$z = \$a * 3 - 2</code>	3	?	7
3	<code>if (\$z >= 1)</code>	true		
4	<code>\$y = 6 * \$a</code>	3	18	7
5	<code>echo \$z, ", ", \$y</code>	It displays: 7 18		

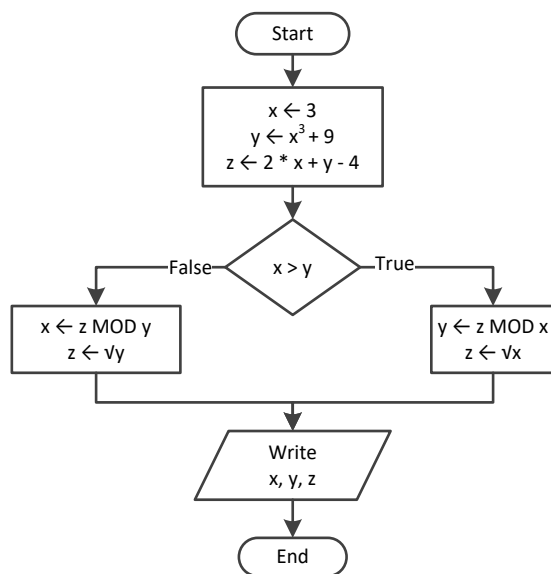
For input value of 0.5

Step	Statement	\$a	\$y	\$z
1	<code>\$a = trim(fgets(STDIN))</code>	0.5	?	?
2	<code>\$z = \$a * 3 - 2</code>	0.5	?	-0.5
3	<code>if (\$z >= 1)</code>	false		
4	<code>\$z++</code>	0.5	?	0.5
5	<code>\$y = 6 * \$a + \$z</code>	0.5	3.5	0.5
6	<code>echo \$z, ", ", \$y</code>	It displays: 0.5, 3.5		



2. Solution

Step	Statement	\$x	\$y	\$z
1	$\$x = 3$	3	?	?
2	$\$y = \$x ** 3 + 9$	3	36	?
3	$\$z = 2 * \$x + \$y - 4$	3	36	38
4	if ($\$x > \y)	false		
5	$\$x = \$z \% \$y$	2	36	38
6	$\$z = \text{sqrt}(\$y)$	2	36	6
7	echo $\$x, ", ", \$y, ", ", \$z$	It displays: 2, 36, 6		



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	<code>\$x = trim(fgets(STDIN))</code>	10	?	?	?
2	<code>\$w = \$x * 3 - 15</code>	10	?	15	?
3	<code>\$z = (\$w + 7) * (\$x + 4) - 10</code>	10	?	15	298
4	<code>if (\$w > \$x && \$z > \$x)</code>	true			
5	<code>\$x++</code>	11	?	15	298
6	<code>\$y = \$x / 2 + 4</code>	11	9.5	15	298
7	<code>echo \$y</code>	It displays: 9.5			

For input value of 2

Step	Statement	\$x	\$y	\$w	\$z
1	<code>\$x = trim(fgets(STDIN))</code>	2	?	?	?
2	<code>\$w = \$x * 3 - 15</code>	2	?	-9	?
3	<code>\$z = (\$w + 7) * (\$x + 4) - 10</code>	2	?	-9	-22
4	<code>if (\$w > \$x && \$z > \$x)</code>	false			
5	<code>\$y = \$x / 4 + 2</code>	2	2.5	-9	-22
6	<code>echo \$y</code>	It displays: 2.5			

4. Solution

```
<?php
    echo "Enter team name 1: ";
    $name1 = trim(fgets(STDIN));
    echo "Enter team name 2: ";
    $name2 = trim(fgets(STDIN));

    echo "Enter goals ", $name1, " scored: ";
    $goals1 = trim(fgets(STDIN));
    echo "Enter goals ", $name2, " scored: ";
    $goals2 = trim(fgets(STDIN));
```

```

    if ($goals1 > $goals2) {
        echo "Winner: ", $name1, "\n";
    }
    else {
        echo "Winner: ", $name2, "\n";
    }
}
?>

```

5. Solution

```

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

```

6. 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\n";
}
else {
    echo $x, " is neither a multiple of 6 nor a multiple of 7\n";
}
?>

```

7. 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: ", $x, " = ", (int)($x / 4), " x 4 + ", $y, "\n";
?>

```

8. Solution

```

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

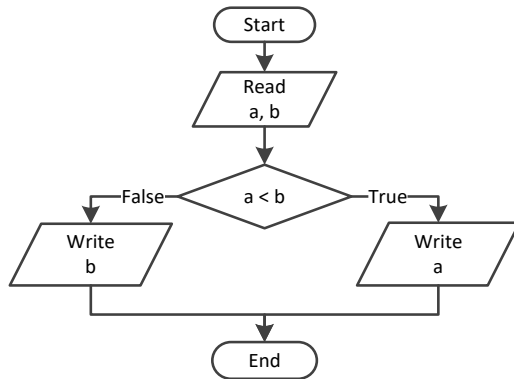
```

```

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

```

9. Solution



```

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

if ($a < $b) {
    echo $a, "\n";
}
else {
    echo $b, "\n";
}
?>

```

10. 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\n";
}
else {
    echo "Given numbers cannot be lengths of the three sides of a triangle\n";
}
?>

```

11. Solution

```

<?php

```



```

$a = trim(fgets(STDIN));
$b = trim(fgets(STDIN));
$c = trim(fgets(STDIN));

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

```

12. Solution

```

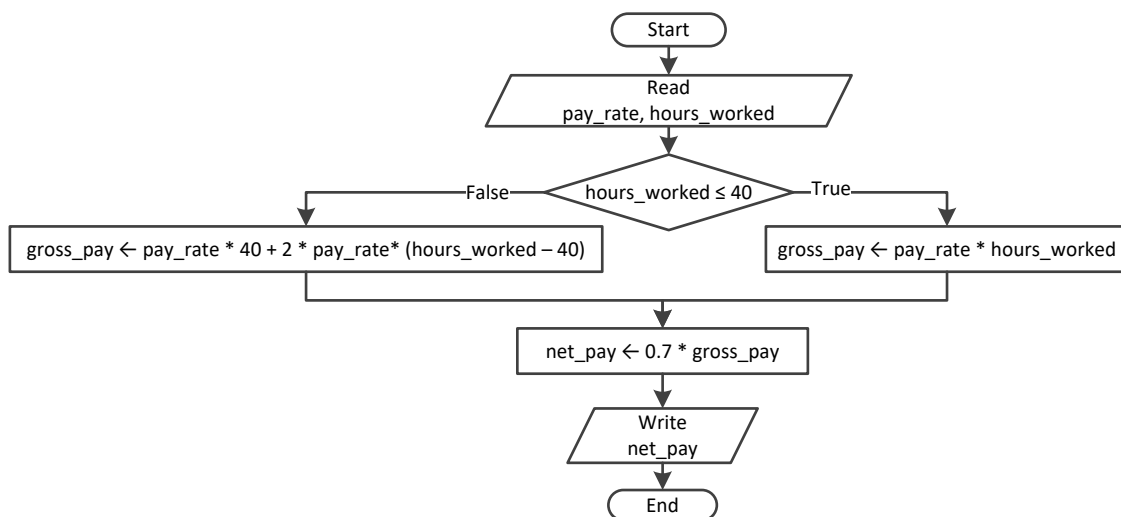
<?php
echo "Enter 1st jump in meters: ";
$a = trim(fgets(STDIN));
echo "Enter 2nd jump in meters: ";
$b = trim(fgets(STDIN));
echo "Enter 3rd jump in meters: ";
$c = trim(fgets(STDIN));

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

if ($average >= 8) {
    echo "Qualified\n";
}
else {
    echo "Disqualified\n";
}
?>

```

13. Solution



```

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

```

```

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

$net_pay = 0.7 * $gross_pay;
echo $net_pay, "\n";
?>

```

14. 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\n";
}
else {
    $miles_left = 6000 - $r;
    echo "Your car needs a minor service in ", $miles_left, " miles\n";
}
?>

```

15. Solution

```

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

$s1 = 0.5 * $a1 * $t ** 2;
$s2 = 0.5 * $a2 * $t ** 2;

echo "Distance between them: ", abs($s1 - $s2), " meters";

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

```

18.2 Review Questions: True/False

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

Chapter 18

18.3 Review Exercises

1. Solution

For input value of 5

Step	Statement	\$q	\$b
1	<code>\$q = trim(fgets(STDIN))</code>	5	?
2	<code>if (\$q > 0 && \$q <= 50)</code>	true	
3	<code>\$b = 1</code>	5	1
4	<code>echo \$b</code>	It displays: 1	

For input value of 150

Step	Statement	\$q	\$b
1	<code>\$q = trim(fgets(STDIN))</code>	150	?
2	<code>if (\$q > 0 && \$q <= 50)</code>	false	
3	<code>elseif (\$q > 50 && \$q <= 100)</code>	false	
4	<code>elseif (\$q > 100 && \$q <= 200)</code>	true	
5	<code>\$b = 3</code>	150	3
6	<code>echo \$b</code>	It displays: 3	

For input value of 250

Step	Statement	\$q	\$b
1	<code>\$q = trim(fgets(STDIN))</code>	250	?
2	<code>if (\$q > 0 && \$q <= 50)</code>	false	
3	<code>elseif (\$q > 50 && \$q <= 100)</code>	false	
4	<code>elseif (\$q > 100 && \$q <= 200)</code>	false	
5	<code>\$b = 4</code>	250	4
6	<code>echo \$b</code>	It displays: 4	

For input value of -1

Step	Statement	\$q	\$b
1	<code>\$q = trim(fgets(STDIN))</code>	-1	?
2	<code>if (\$q > 0 && \$q <= 50)</code>	false	
3	<code>elseif (\$q > 50 && \$q <= 100)</code>	false	
4	<code>elseif (\$q > 100 && \$q <= 200)</code>	false	

5	<code>\$b = 4</code>	-1	4
6	<code>echo \$b</code>	It displays: 4	

2. Solution

For input value of 5

Step	Statement	\$amount	\$discount	\$payment
1	<code>\$amount = trim(fgets(STDIN))</code>	5	?	?
2	<code>if (\$amount < 20)</code>	true		
3	<code>\$discount = 0</code>	5	0	?
4	<code>\$payment = \$amount - \$amount * \$discount / 100</code>	5	0	5
5	<code>echo \$discount, ", ", \$payment</code>	It displays: 0, 5		

For input value of 150

Step	Statement	\$amount	\$discount	\$payment
1	<code>\$amount = trim(fgets(STDIN))</code>	150	?	?
2	<code>if (\$amount < 20)</code>	false		
3	<code>elseif (\$amount >= 20 && \$amount < 60)</code>	false		
4	<code>elseif (\$amount >= 60 && \$amount < 100)</code>	false		
5	<code>elseif (\$amount >= 100)</code>	true		
6	<code>\$discount = 15</code>	150	15	?
7	<code>\$payment = \$amount - \$amount * \$discount / 100</code>	150	15	127.5
8	<code>echo \$discount, ", ", \$payment</code>	It displays: 15, 127.5		

For input value of -1

Step	Statement	\$amount	\$discount	\$payment
1	<code>\$amount = trim(fgets(STDIN))</code>	-1	?	?
2	<code>if (\$amount < 20)</code>	true		
3	<code>\$discount = 0</code>	-1	0	?
4	<code>\$payment = \$amount - \$amount * \$discount / 100</code>	-1	0	-1
5	<code>echo \$discount, ", ", \$payment</code>	It displays: 0, -1		

3. Solution

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

if ($a < 1) {
    $y = 5 + $a;
    echo $y;
}
elseif ($a < 5) {
    $y = 23 / $a;
    echo $y;
```

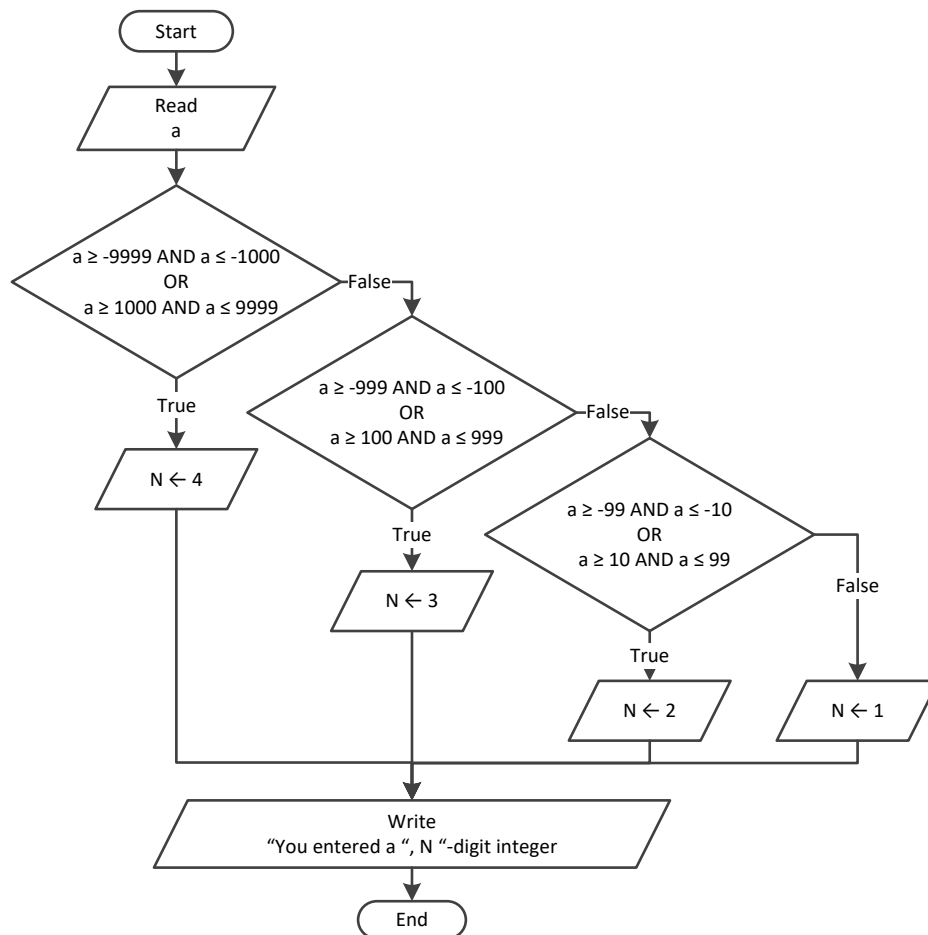
```
}  
elseif ($a < 10) {  
    $y = 5 * $a;  
    echo $y;  
}  
else {  
    echo "Error! ";  
}  
?>
```

4. Solution

```
<?php  
    echo "Enter team name 1: ";  
    $name1 = trim(fgets(STDIN));  
    echo "Enter team name 2: ";  
    $name2 = trim(fgets(STDIN));  
  
    echo "Enter goals ", $name1, " scored: ";  
    $goals1 = trim(fgets(STDIN));  
    echo "Enter goals ", $name2, " scored: ";  
    $goals2 = trim(fgets(STDIN));  
  
    if ($goals1 > $goals2) {  
        echo "Winner: ", $name1, "\n";  
    }  
    elseif ($goals2 > $goals1) {  
        echo "Winner: ", $name2, "\n";  
    }  
    else {  
        echo "It's a tie!\n";  
    }  
?>
```

5. Solution

First approach



```

<?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\n";
?>

```

Second approach

```

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

```

```

$a_string = (string)abs($a);
echo "You entered a ", strlen($a_string), "-digit integer\n";
?>

```

6. Solution

First approach

```

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

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

```

Second approach

```

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

if ($a >= -9999 && $a <= 9999) {
    $a_string = (string)abs($a);
    echo "You entered a ", strlen($a_string), "-digit integer\n";
}
else {
    echo "Error: Invalid value!\n";
}
?>

```

7. 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 = $usd * 0.87;
    echo "$", $usd, " = ", $eur, " EUR\n";
}
elseif ($ch == 2) {
    $gbp = $usd * 0.78;
    echo "$", $usd, " = ", $gbp, " GBP\n";
}
elseif ($ch == 3) {
    $jpy = $usd * 108.55;
    echo "$", $usd, " = ", $jpy, " JPY\n";
}
else {
    $cad = $usd * 1.33;
    echo "$", $usd, " = ", $cad, " CAD\n";
}
?>

```

8. Solution

```

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

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

```

9. Solution

```

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

if ($m < 1 || $m > 12) {
    echo "Error: Invalid value!\n";
}
elseif ($m <= 2 || $m == 12) {
    echo "Winter\n";
}
elseif ($m <= 5) {

```



```
        echo "Spring\n";
    }
    elseif ($m <= 8) {
        echo "Summer\n";
    }
    else {
        echo "Fall (Autumn)\n";
    }
?>
```

10. Solution

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

    $x = (int)($n);
    $y = (int)($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 ";

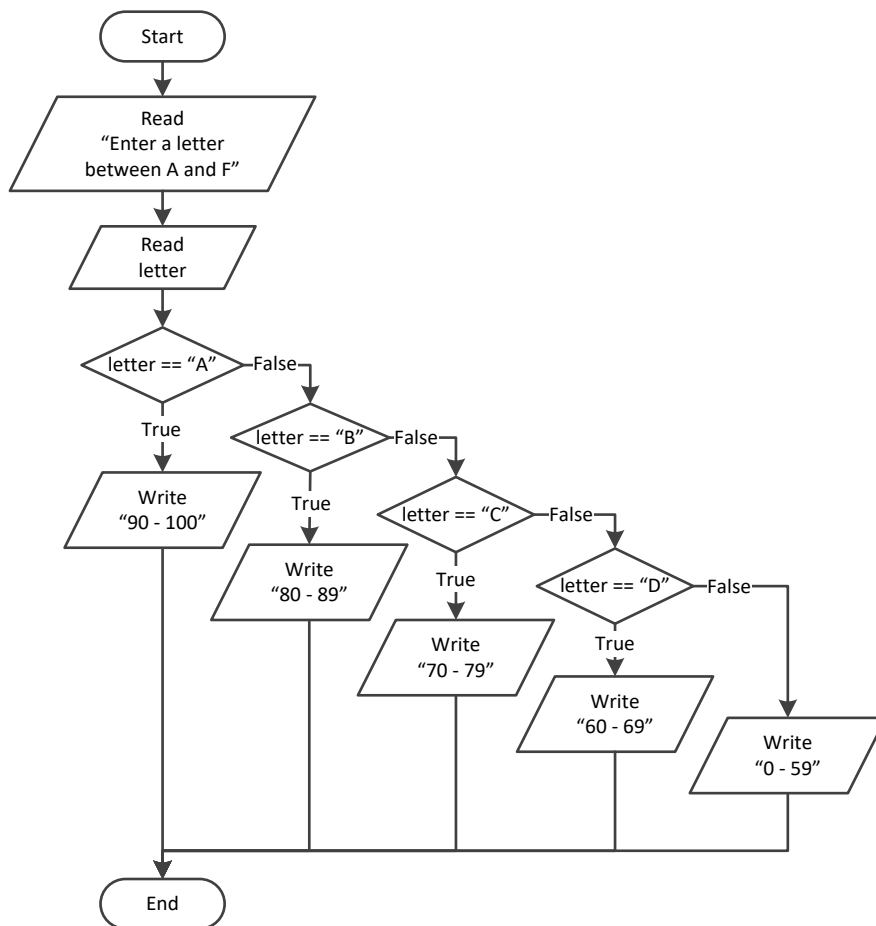
    if ($y == 1) {
        echo "one\n";
    }
    elseif ($y == 2) {
        echo "two\n";
    }
    elseif ($y == 3) {
        echo "three\n";
    }
    elseif ($y == 4) {
        echo "four\n";
    }
    elseif ($y == 5) {
        echo "five\n";
    }
    elseif ($y == 6) {
        echo "six\n";
    }
}
```

```

elseif ($y == 7) {
    echo "seven\n";
}
elseif ($y == 8) {
    echo "eight\n";
}
elseif ($y == 9) {
    echo "nine\n";
}
elseif ($y == 0) {
    echo "zero\n";
}
?>

```

11. Solution



```

<?php
echo "Enter a letter between A and F: ";
$letter = trim(fgets(STDIN));

if ($letter == "A") {
    echo "90 - 100\n";
}
elseif ($letter == "B") {
    echo "80 - 89\n";
}

```

```
}  
elseif ($letter == "C") {  
    echo "70 - 79\n";  
}  
elseif ($letter == "D") {  
    echo "60 - 69\n";  
}  
else {  
    echo "0 - 59\n";  
}  
?>
```

19.2 Review Questions: True/False

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

Chapter 19

19.3 Review Exercises

1. Solution

For input value of 1

Step	Statement	\$a	\$x	\$y
1	<code>\$a = trim(fgets(STDIN))</code>	1	?	?
2	<code>\$x = 0</code>	1	0	?
3	<code>\$y = 0</code>	1	0	0
4	<code>case \$a == 1</code>	true		
5	<code>\$x = \$x + 5</code>	1	5	0
6	<code>\$y = \$y + 5</code>	1	5	5
7	<code>echo \$x, ", ", \$y</code>	It displays: 5, 5		

For input value of 3

Step	Statement	\$a	\$x	\$y
1	<code>\$a = trim(fgets(STDIN))</code>	3	?	?
2	<code>\$x = 0</code>	3	0	?
3	<code>\$y = 0</code>	3	0	0
4	<code>case \$a == 1</code>	false		
5	<code>case \$a == 2</code>	false		
6	<code>case \$a == 3</code>	true		
7	<code>\$x = \$x - 9</code>	3	-9	0
8	<code>\$y = \$y + 3</code>	3	-9	3
9	<code>echo \$x, ", ", \$y</code>	It displays: -9, 3		

For input value of 250

Step	Statement	\$a	\$x	\$y
1	<code>\$a = trim(fgets(STDIN))</code>	250	?	?
2	<code>\$x = 0</code>	250	0	?
3	<code>\$y = 0</code>	250	0	0
4	<code>case \$a == 1</code>	false		
5	<code>case \$a == 2</code>	false		
6	<code>case \$a == 3</code>	false		

7	<code>\$x = \$x + 3</code>	250	3	0
8	<code>\$y++</code>	250	3	1
9	<code>echo \$x, ", ", \$y</code>	It displays: 3, 1		

2. Solution

For input values of 10, 2, 5

Step	Statement	\$a	\$x	\$y
1	<code>\$a = trim(fgets(STDIN))</code>	10	?	?
2	<code>\$x = trim(fgets(STDIN))</code>	10	2	?
3	<code>\$y = trim(fgets(STDIN))</code>	10	2	5
4	<code>case \$a == 10</code>	true		
5	<code>\$x = \$x % 2</code>	10	0	5
6	<code>\$y = \$y ** 2</code>	10	0	25
7	<code>echo \$x, ", ", \$y</code>	It displays: 0, 25		

For input values of 5, 2, 3

Step	Statement	\$a	\$x	\$y
1	<code>\$a = trim(fgets(STDIN))</code>	5	?	?
2	<code>\$x = trim(fgets(STDIN))</code>	5	2	?
3	<code>\$y = trim(fgets(STDIN))</code>	5	2	3
4	<code>case \$a == 10</code>	false		
5	<code>case \$a == 3</code>	false		
6	<code>case \$a == 5</code>	true		
7	<code>\$x = \$x + 4</code>	5	6	3
8	<code>\$y += 7</code>	5	6	10
9	<code>echo \$x, ", ", \$y</code>	It displays: 6, 10		

For input values of 4, 6, 2

Step	Statement	\$a	\$x	\$y
1	<code>\$a = trim(fgets(STDIN))</code>	4	?	?
2	<code>\$x = trim(fgets(STDIN))</code>	4	6	?
3	<code>\$y = trim(fgets(STDIN))</code>	4	6	2
4	<code>case \$a == 10</code>	false		
5	<code>case \$a == 3</code>	false		
6	<code>case \$a == 5</code>	false		
7	<code>\$x -= 3</code>	4	3	2.0
8	<code>\$y++</code>	4	3	3.0
9	<code>echo \$x, ", ", \$y</code>	It displays: 3, 3		

3. Solution

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

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

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

5. Solution

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

switch ($roman) {
    case "I":
        echo "1\n";
        break;
    case "II":
        echo "2\n";
        break;
    case "III":
        echo "3\n";
        break;
    case "IV":
        echo "4\n";
        break;
    case "V":
        echo "5\n";
```

```
        break;
    case "VI":
        echo "6\n";
        break;
    case "VII":
        echo "7\n";
        break;
    case "VIII":
        echo "8\n";
        break;
    case "IX":
        echo "9\n";
        break;
    case "X":
        echo "10\n";
        break;
    default:
        echo "Error\n";
    }
?>
```

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\n";
        break;
    case 2:
        echo "You are awarded 10 points\n";
        break;
    case 3:
        echo "You are awarded 20 points\n";
        break;
    default:
        echo "You are awarded 45 points\n";
    }
?>
```

7. Solution

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

$i = rand(0, 2);

switch ($i) {
    case 0:
```



```
        echo "Good morning ", $name, "\n";
        break;
    case 1:
        echo "Good evening ", $name, "\n";
        break;
    case 2:
        echo "Good night ", $name, "\n";
        break;
    }
?>
```

8. Solution

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

switch ($num) {
    case "zero":
        echo 0, "\n";
        break;
    case "one":
        echo 1, "\n";
        break;
    case "two":
        echo 2, "\n";
        break;
    case "three":
        echo 3, "\n";
        break;
    case "four":
        echo 4, "\n";
        break;
    case "five":
        echo 5, "\n";
        break;
    case "six":
        echo 6, "\n";
        break;
    case "seven":
        echo 7, "\n";
        break;
    case "eight":
        echo 8, "\n";
        break;
    case "nine":
        echo 9, "\n";
        break;
    default:
        echo "I don't know this number!\n";
}
?>
```

9. Solution

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

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

20.3 Review Questions: True/False

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

Chapter 20

20.4 Review Exercises

1. Solution

For input values of 20, 1

Step	Statement	\$x	\$y
1	<code>\$x = trim(fgets(STDIN))</code>	20	?
2	<code>\$y = trim(fgets(STDIN))</code>	20	1
3	<code>if (\$x < 30)</code>	true	
4	<code>case \$y == 1</code>	true	
5	<code>\$x = \$x % 3</code>	2	1
6	<code>\$y = 5</code>	2	5
7	<code>echo \$x, ", ", \$y</code>	It displays: 2, 5	

For input values of 20, 3

Step	Statement	\$x	\$y
1	<code>\$x = trim(fgets(STDIN))</code>	20	?
2	<code>\$y = trim(fgets(STDIN))</code>	20	3
3	<code>if (\$x < 30)</code>	true	
4	<code>case \$y == 1</code>	false	
5	<code>case \$y == 2</code>	false	
6	<code>case \$y == 3</code>	true	
7	<code>\$x = \$x + 5</code>	25	3
8	<code>\$y += 3</code>	25	6
9	<code>echo \$x, ", ", \$y</code>	It displays: 25, 6	

For input values of 12, 8

Step	Statement	\$x	\$y
1	<code>\$x = trim(fgets(STDIN))</code>	12	?
2	<code>\$y = trim(fgets(STDIN))</code>	12	8
3	<code>if (\$x < 30)</code>	true	
4	<code>case \$y == 1</code>	false	
5	<code>case \$y == 2</code>	false	
6	<code>case \$y == 3</code>	false	
7	<code>\$x -= 2</code>	10	8

8	<code>\$y++</code>	10	9
9	<code>echo \$x, ", ", \$y</code>	It displays: 10, 9	

For input values of 50, 0

Step	Statement	\$x	\$y
1	<code>\$x = trim(fgets(STDIN))</code>	50	?
2	<code>\$y = trim(fgets(STDIN))</code>	50	0
3	<code>if (\$x < 30)</code>	false	
4	<code>\$y++</code>	50	1
5	<code>echo \$x, ", ", \$y</code>	It displays: 50, 1	

2. Solution

For input values of 60, 25

Step	Statement	\$x	\$y
1	<code>\$x = trim(fgets(STDIN))</code>	60	?
2	<code>\$y = trim(fgets(STDIN))</code>	60	25
3	<code>if ((\$x + \$y) / 2 <= 20)</code>	false	
4	<code>if (\$y < 15)</code>	false	
5	<code>elseif (\$y < 23)</code>	false	
6	<code>\$x = 2 * \$x + 5</code>	125	25
7	<code>\$y += 1</code>	125	26
8	<code>echo \$x, ", ", \$y</code>	It displays: 125, 26	

For input values of 50, 8

Step	Statement	\$x	\$y
1	<code>\$x = trim(fgets(STDIN))</code>	50	?
2	<code>\$y = trim(fgets(STDIN))</code>	50	8
3	<code>if ((\$x + \$y) / 2 <= 20)</code>	false	
4	<code>if (\$y < 15)</code>	true	
5	<code>\$x = \$x % 4</code>	2	8
6	<code>\$y = 2</code>	2	2
7	<code>echo \$x, ", ", \$y</code>	It displays: 2, 2	

For input values of 20, 15

Step	Statement	\$x	\$y
1	<code>\$x = trim(fgets(STDIN))</code>	20	?
2	<code>\$y = trim(fgets(STDIN))</code>	20	15
3	<code>if ((\$x + \$y) / 2 <= 20)</code>	true	
4	<code>if (\$y < 10)</code>	false	
5	<code>elseif (\$y < 20)</code>	true	

6	<code>\$x = \$x * 5</code>	100	15
7	<code>\$y += 2</code>	100	17
8	<code>echo \$x, ", ", \$y</code>	It displays: 100, 17	

For input values of 10, 30

Step	Statement	\$x	\$y
1	<code>\$x = trim(fgets(STDIN))</code>	10	?
2	<code>\$y = trim(fgets(STDIN))</code>	10	30
3	<code>if ((\$x + \$y) / 2 <= 20)</code>	true	
4	<code>if (\$y < 10)</code>	false	
5	<code>elseif (\$y < 20)</code>	false	
6	<code>\$x = \$x - 2</code>	8	30
7	<code>\$y += 3</code>	8	33
8	<code>echo \$x, ", ", \$y</code>	It displays: 8, 33	

3. Solution

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

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

4. Solution

```
<?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\n";
}
```

```

    }
    else {
        if ($a == $b && $b == $c) {
            echo "Equilateral\n";
        }
        elseif ($a ** 2 == $b ** 2 + $c ** 2 || $b ** 2 == $a ** 2 + $c ** 2 || $c ** 2 == $a ** 2 + $b ** 2) {
            echo "Right (or right-angled)\n";
        }
        else {
            echo "Not special\n";
        }
    }
}
?>

```

5. 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!\n";
}
else {
    echo "Enter the amount of money (an integer value) that you want to withdraw: ";
    $amount = trim(fgets(STDIN));
    $usd10 = (int)($amount / 10);
    $r = $amount % 10;
    $usd5 = (int)($r / 5);
    $usd1 = $r % 5;
    echo $usd10, " note(s) of $10 ", $usd5, " note(s) of $5 ";
    echo "and ", $usd1, " note(s) of $1\n";
}
?>

```

6. 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\n";
    }
    else {
        echo "The day is hot and not windy\n";
    }
}
else {
    if ($w > 12) {
        echo "The day is cold and windy\n";
    }
    else {
        echo "The day is cold and not windy\n";
    }
}
?>
```

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";
}

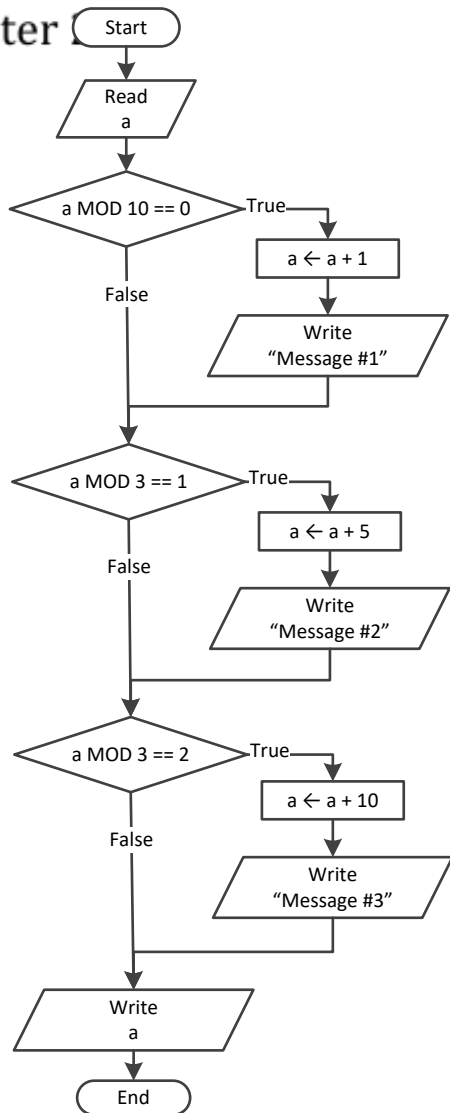
if ($w > 12) {
    $message2 = "windy";
}
else {
    $message2 = "not windy";
}

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

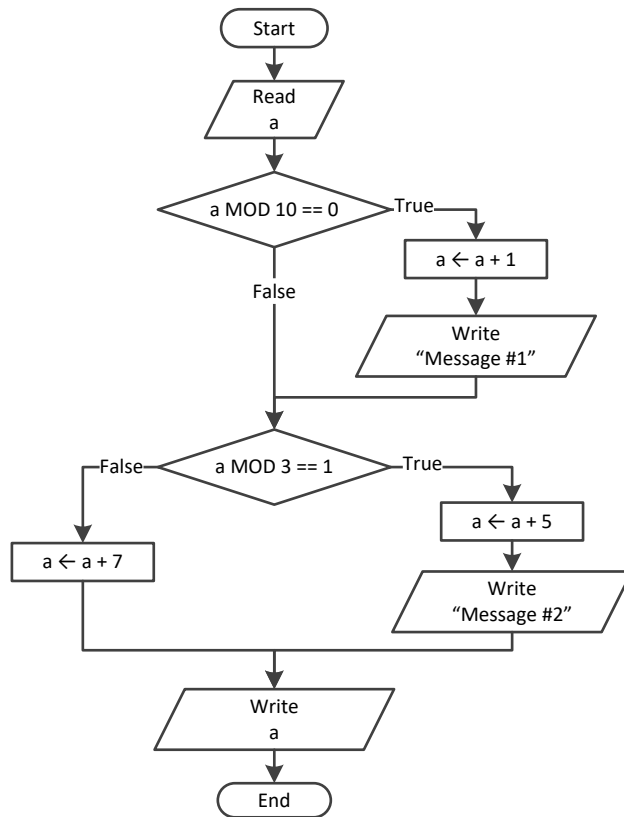
21.4 Review Exercises

1. Solution

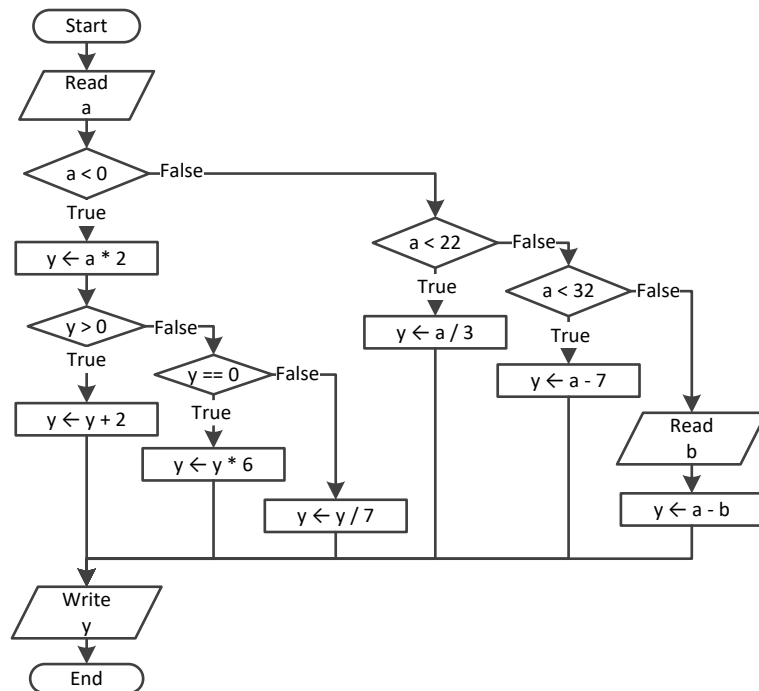
Chapter



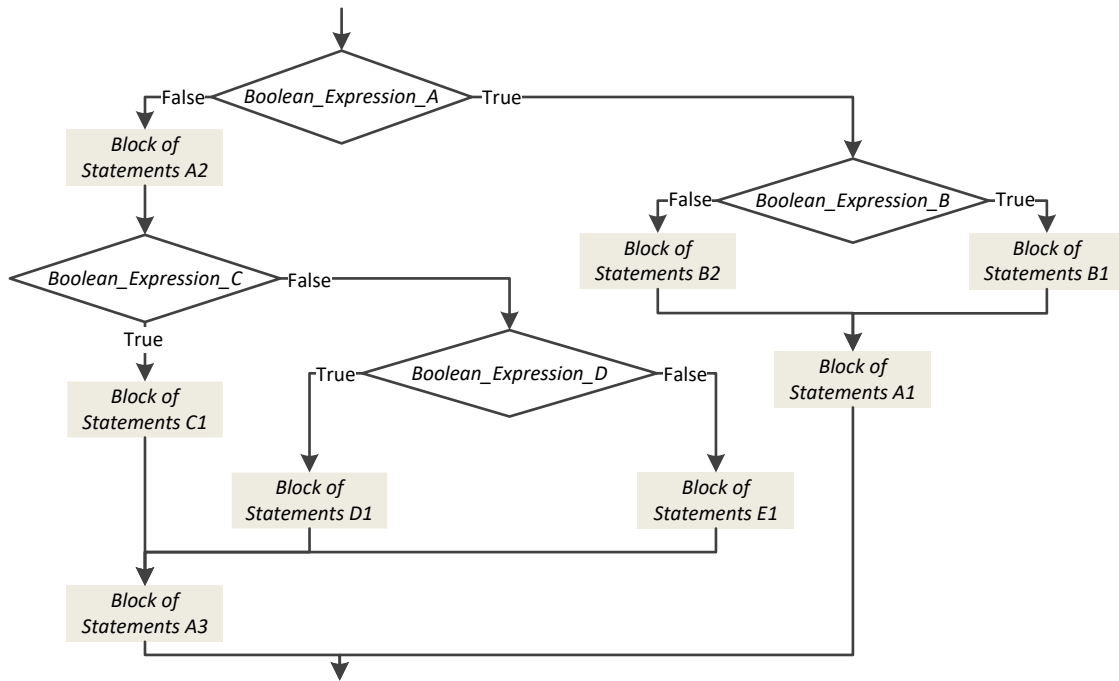
2. Solution



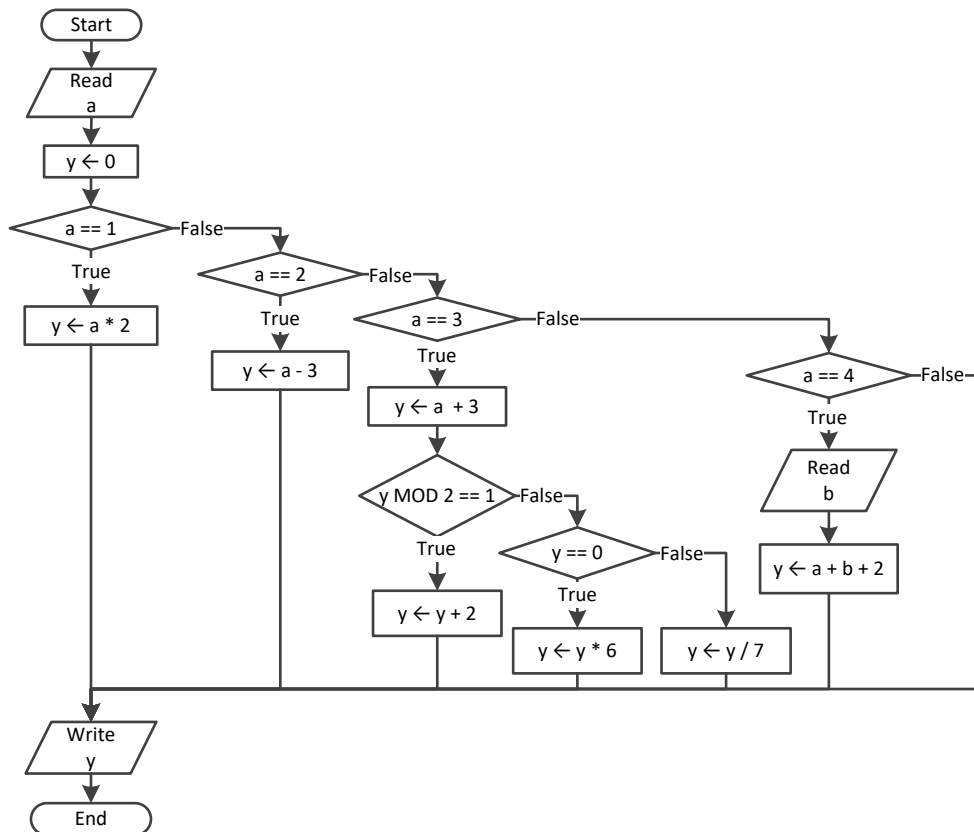
3. Solution



4. Solution



5. Solution



6. Solution

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

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

8. Solution

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

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

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

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

echo $y;
?>
```

9. Solution

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

if ($x > 0) {
    if ($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 == -1) {
        echo "Bye";
    }
    else {
        echo "Invalid Number";
    }
}
?>
```

10. 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;
}
}
?>
```

11. 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;
?>
```

22.9 Review Questions: True/False

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

Chapter 22**22.10 Review Questions: Multiple Choice**

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

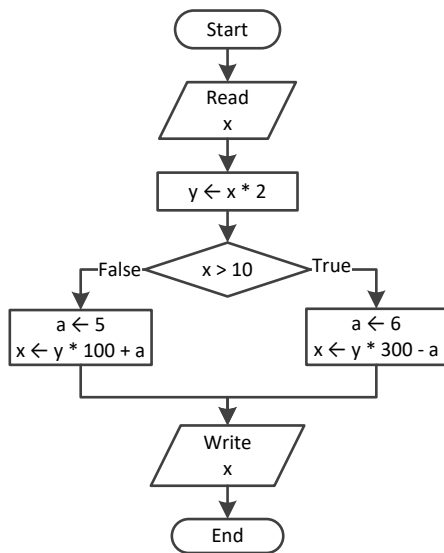
22.11 Review Exercises

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
    $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
    $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 variable `$a` is less than or equal to 10. The correct program 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
$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
$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
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";  
    }  
?>
```

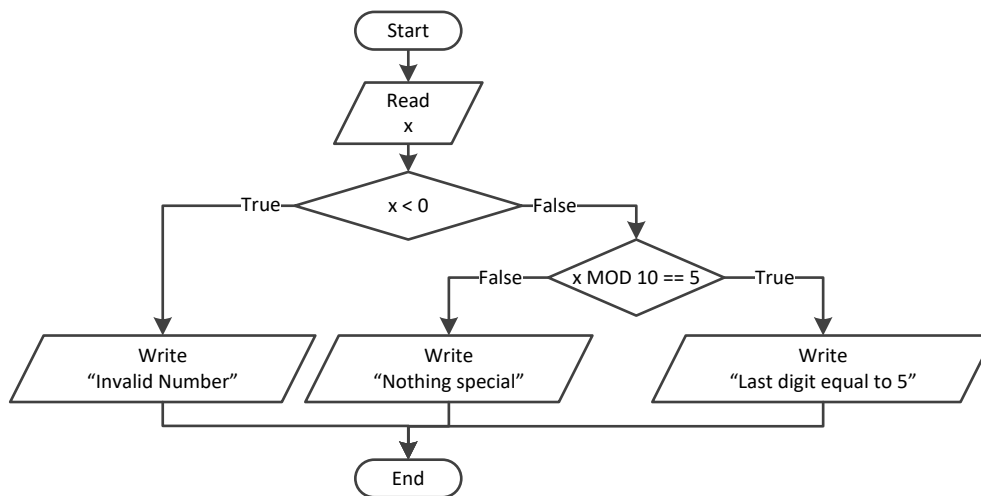
23.6 Review Exercises

1. Solution

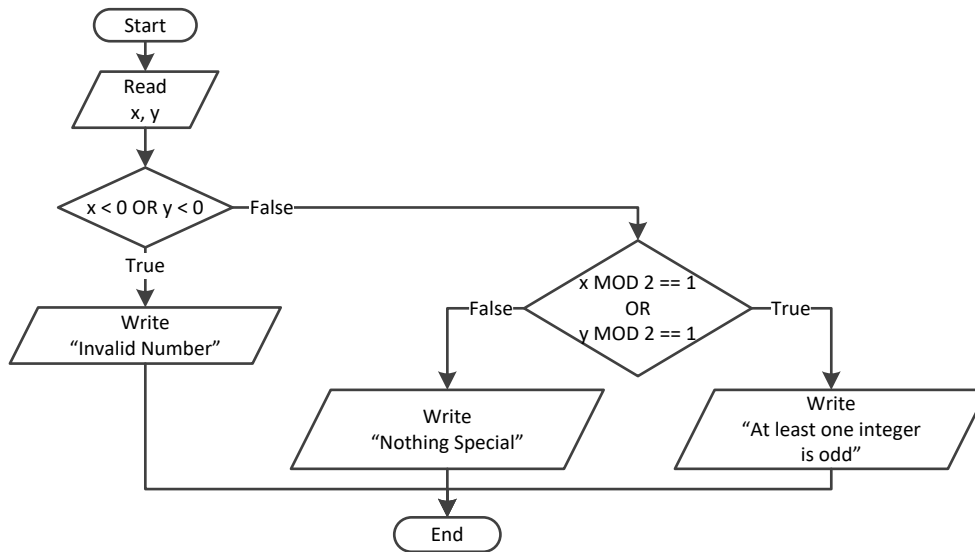
Chapter 23

```
<?php
echo "Enter a non-negative number: ";
$x = trim(fgets(STDIN));
if ($x < 0) {
    echo "Error! You entered a negative value\n";
}
else {
    echo "The square root of ", $x, " is ", sqrt($x), "\n";
}
?>
```

2. Solution



3. Solution



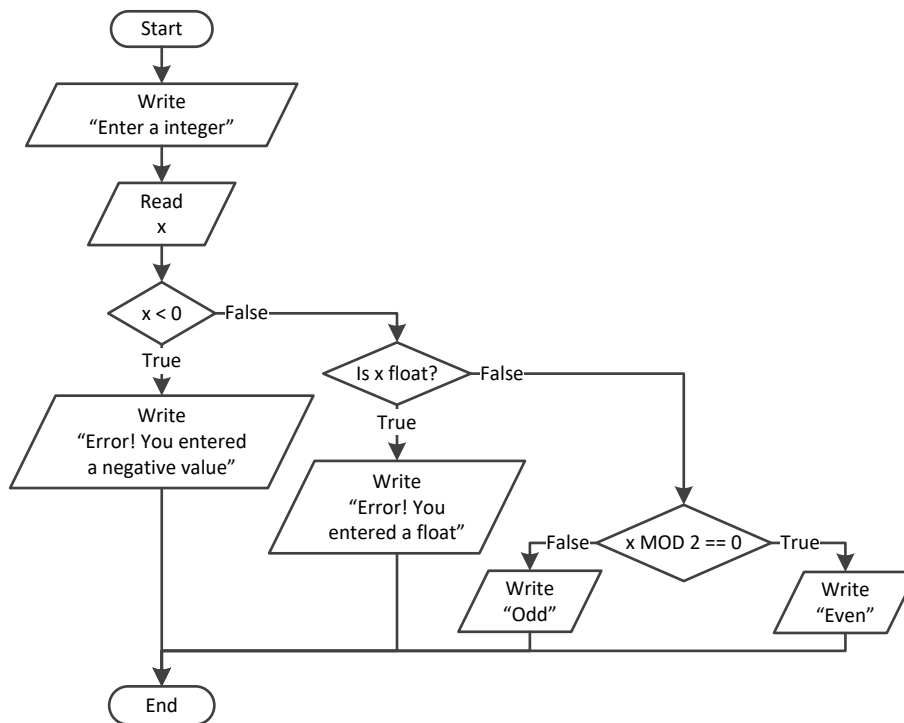
```

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

if (x < 0 || y < 0) {
    echo "Invalid Number\n";
}
else {
    if (x % 2 == 1 || y % 2 == 1) {
        echo "At least one integer is odd\n";
    }
    else {
        echo "Nothing Special\n";
    }
}
?>

```

4. Solution

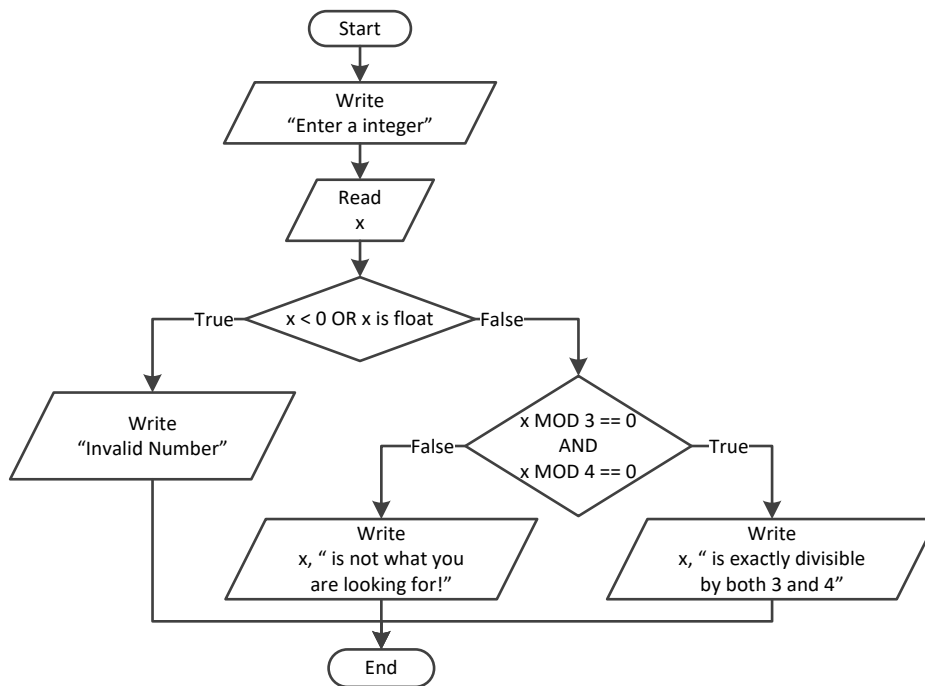


```

<?php
echo "Enter a non-negative number: ";
$x = trim(fgets(STDIN));
if ($x < 0) {
    echo "Error! You entered a negative value\n";
}
elseif ($x != (int)$x) {
    echo "Error! You entered a float\n";
}
elseif ($x % 2 == 0) {
    echo "Even\n";
}
else {
    echo "Odd\n";
}
?>

```

5. Solution



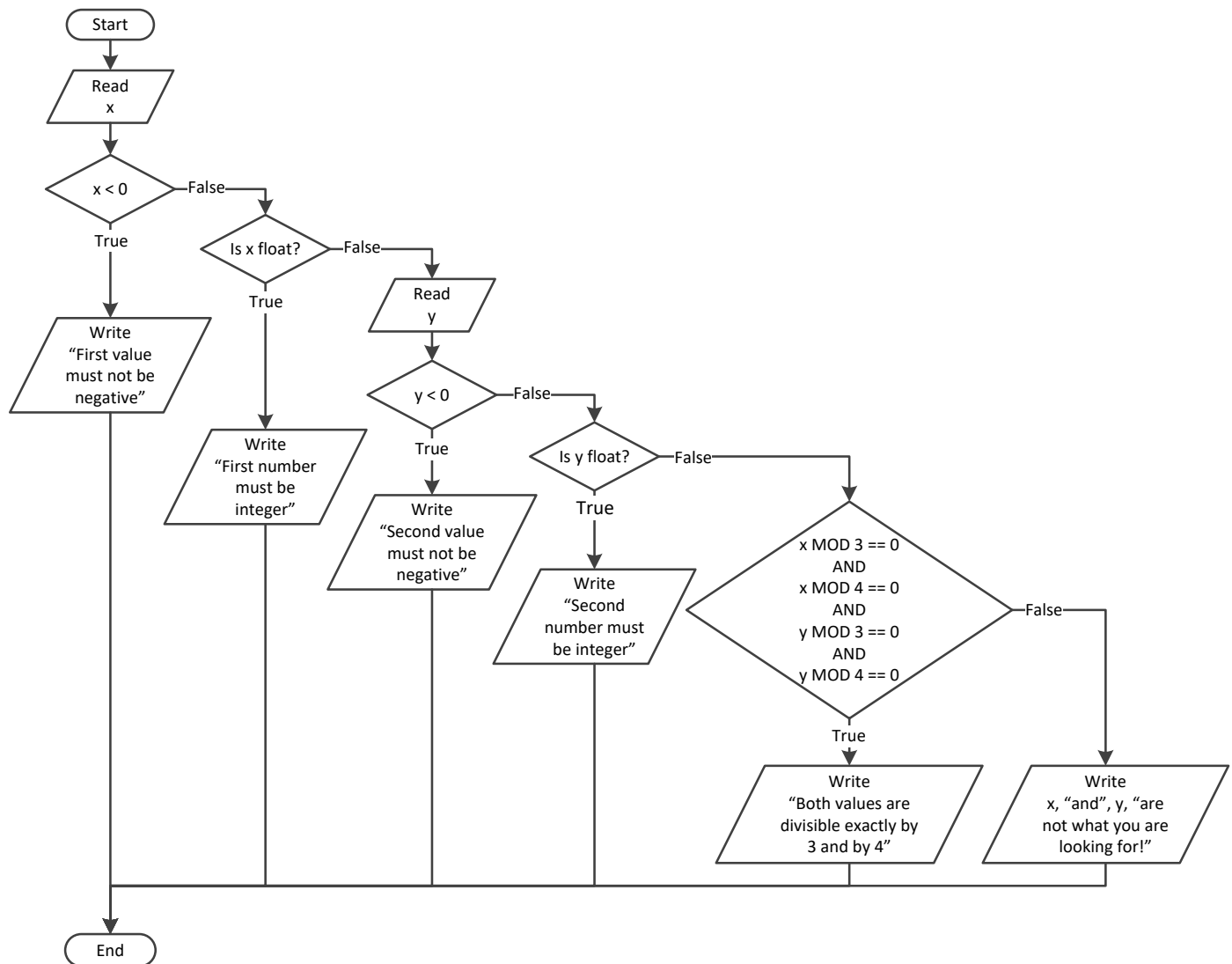
```

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

if ($x < 0 || $x != (int)$x) {
    echo "Invalid Number\n";
}
elseif ($x % 3 == 0 && $x % 4 == 0) {
    echo $x, " is exactly divisible by both 3 and 4\n";
}
else {
    echo $x, " is not what you are looking for!\n";
}
?>

```

6. Solution



```

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

    if ($x < 0) {
        echo "First value must be not be negative\n";
    }
    else {
        if ($x != (int)$x) {
            echo "First number must be integer\n";
        }
        else {
            $y = trim(fgets(STDIN));
            if ($y < 0) {
                echo "Second value must be not be negative\n";
            }
            else {
                if ($y != (int)$y) {

```

```

        echo "Second number must be integer\n";
    }
    else {
        if ($x % 3 == 0 && $x % 4 == 0 && $y % 3 == 0 && $y % 4 == 0 ) {
            echo "Both values are divisible exactly by 3 and by 4\n";
        }
        else {
            echo "Nothing Special\n";
        }
    }
}
}
?>

```

7. 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 || $choice > 4) {
    echo "Wrong choice\n";
}
else {
    switch ($choice) {
        case 1:
            if ($t < 0) { //Absolute zero in Kelvin
                echo "Wrong temperature\n";
            }
            else {
                echo 1.8 * $t - 459.67, "\n";
            }
            break;
        case 2:
            if ($t < -459.67) { //Absolute zero in Fahrenheit
                echo "Wrong temperature\n";
            }
            else {
                echo ($t + 459.57) / 1.8, "\n";
            }
            break;
        case 3:
            if ($t < -459.67) { //Absolute zero in Fahrenheit
                echo "Wrong temperature\n";
            }
            else {
                echo ($t * 1.8 + 32), "\n";
            }
            break;
        case 4:
            if ($t < -273.15) { //Absolute zero in Celsius
                echo "Wrong temperature\n";
            }
            else {
                echo ($t * 9/5 + 32), "\n";
            }
            break;
    }
}

```



```

    }
    else {
        echo 5 / 9 * ($t - 32), "\n";
    }
    break;
case 4:
    if ($t < -273.15) { //Absolute zero in Celcius
        echo "Wrong temperature\n";
    }
    else {
        echo 9 / 5 * $t + 32, "\n";
    }
    break;
}
}
?>

```

8. 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, "\n";
        break;
    case "-":
        echo $a - $b, "\n";
        break;
    case "*":
        echo $a * $b, "\n";
        break;
    case "/":
        if ($b == 0) {
            echo "Error: Division by zero\n";
        }
        else {
            echo $a / $b, "\n";
        }
        break;
    case "DIV":
        if ($b == 0) {
            echo "Error: Division by zero\n";
        }
        else {
            echo (int)($a / $b), "\n";
        }
    }
}

```

```

    }
    break;
case "MOD":
    if ($b == 0) {
        echo "Error: Division by zero\n";
    }
    else {
        echo $a % $b, "\n";
    }
    break;
case "POWER":
    echo $a ** $b, "\n";
    break;
}
?>

```

9. 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, "\n";
        break;
    case "-":
        echo $a - $b, "\n";
        break;
    case "*":
        echo $a * $b, "\n";
        break;
    case "/":
        if ($b == 0) {
            echo "Error: Division by zero\n";
        }
        else {
            echo $a / $b, "\n";
        }
        break;
    case "DIV":
        if ($b == 0) {
            echo "Error: Division by zero\n";
        }
        else {
            echo (int)($a / $b), "\n";
        }
}

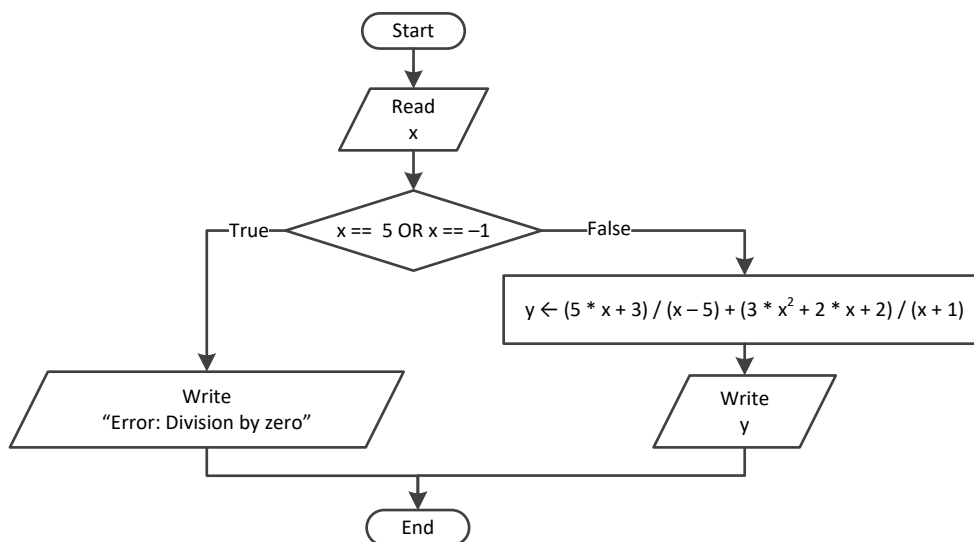
```

```

    break;
case "MOD":
    if ($b == 0) {
        echo "Error: Division by zero\n";
    }
    else {
        echo $a % $b, "\n";
    }
    break;
case "POWER":
    echo $a ** $b, "\n";
    break;
default:
    echo "Error: Invalid operator\n";
}
?>

```

10. Solution



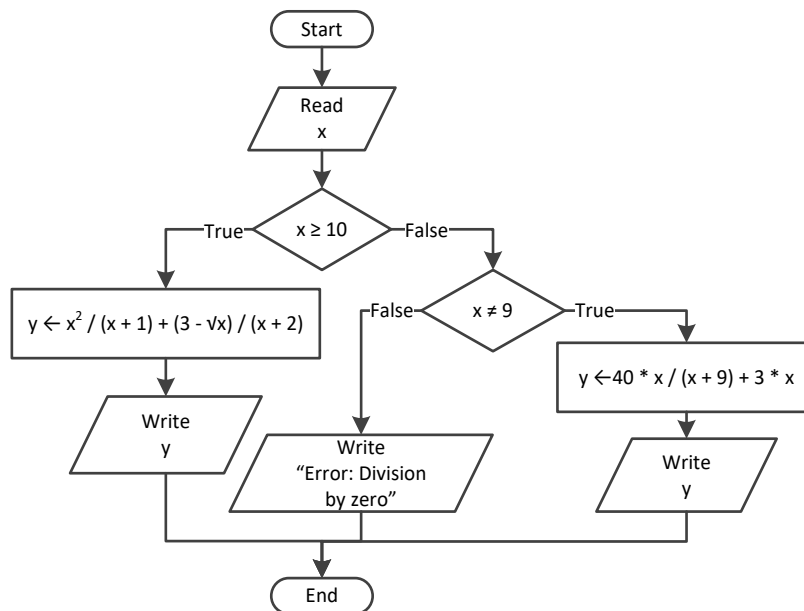
```

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

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

```

11. Solution



```

<?php
$x = trim(fgets(STDIN));
if ($x >= 10) {
    $y = $x ** 2 / ($x + 1) + (3 - sqrt($x)) / ($x + 2);
    echo $y, "\n";
}
elseif ($x != 9) {
    $y = 40 * $x / ($x + 9) + 3 * $x;
    echo $y, "\n";
}
else {
    echo "Error: Division by zero\n";
}
?>

```

12. Solution

```

<?php
$x = trim(fgets(STDIN));
if ($x <= -15 || $x > 25) {
    $y = $x - 1;
    echo $y, "\n";
}
elseif ($x <= -10) {
    $y = $x / sqrt($x + 30) + (8 + $x) ** 2 / ($x + 1);
    echo $y, "\n";
}
elseif ($x <= 0) {
    $y = abs(40 * $x) / ($x - 8);
    echo $y, "\n";
}

```

```

else {
    if ($x == 9) {
        echo "Error: Division by zero\n";
    }
    elseif ($x < 9) {
        echo "Error: Invalid square root\n";
    }
    else {
        $y = 3 * $x / sqrt($x - 9);
        echo $y, "\n";
    }
}
?>

```

13. 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));

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

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

```

```

    echo $min_name, " ", $max_name, "\n";
?>

```

14. 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));

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

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

    $middle = $age1 + $age2 + $age3 - $minimum - $maximum;
    echo $middle, "\n";
?>

```

15. 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));

    $minimum = $a1;
    $min_name = $n1;
    if ($a2 < $minimum) {

```

```

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

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

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

if (abs($maximum - $middle) < abs($minimum - $middle)) {
    echo $max_name, "\n";
}
else {
    echo $min_name, "\n";
}
?>

```

16. Solution

```

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

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

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

    $total = (int)($digit1 ** 3 + $digit2 ** 3 + $digit3 ** 3);

    if ($total == $x) {
        echo "You entered an Armstrong number!\n";
    }
}

```

```

        else {
            echo "You entered a non-Armstrong number!\n";
        }
    }
    ?>

```

17. 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, "\n";
        }
        else {
            echo 28 - $d, "\n";
        }
    }
    elseif ($m == 4 || $m == 6 || $m == 9 || $m == 11) {
        echo 30 - $d, "\n";
    }
    else {
        echo 31 - $d, "\n";
    }
    ?>

```

18. Solution

First approach

```

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

    $word1 = strtoupper(substr($word, 0, 1)) .
        strtolower(substr($word, 1, 1)) .
        strtoupper(substr($word, 2, 1)) .
        strtolower(substr($word, 3, 1)) .
        strtoupper(substr($word, 4, 1)) .
        strtolower(substr($word, 5, 1));

    $word2 = strtolower(substr($word, 0, 1)) .
        strtoupper(substr($word, 1, 1)) .
        strtolower(substr($word, 2, 1)) .
        strtoupper(substr($word, 3, 1)) .
        strtolower(substr($word, 4, 1)) .
        strtoupper(substr($word, 5, 1));

```



```

    if ($word == $word1 || $word == $word2) {
        echo "Word is okay!\n";
    }
    else {
        echo "Word is not okay\n";
    }
}
?>

```

Second approach

```

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

    $word1 = strtoupper($word[0]) .
        strtolower($word[1]) .
        strtoupper($word[2]) .
        strtolower($word[3]) .
        strtoupper($word[4]) .
        strtolower($word[5]);

    $word2 = strtolower($word[0]) .
        strtoupper($word[1]) .
        strtolower($word[2]) .
        strtoupper($word[3]) .
        strtolower($word[4]) .
        strtoupper($word[5]);

    if ($word == $word1 || $word == $word2) {
        echo "Word is okay!\n";
    }
    else {
        echo "Word is not okay\n";
    }
}
?>

```

19. Solution

```

<?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 / 100.0;

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

```

20. Solution

```

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

    echo "Enter a before-tax amount: ";
    $amount = trim(fgets(STDIN));

    if ($amount < 0) {
        echo "Error! You entered a negative value\n";
    }
    else {
        if ($amount < 50) {
            $discount = 0;
        }
        elseif ($amount < 100) {
            $discount = 1;
        }
        elseif ($amount < 250) {
            $discount = 2;
        }
        else {
            $discount = 3;
        }

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

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

```

21. Solution

```

<?php
    echo "Enter age: ";
    $a = trim(fgets(STDIN));
    if ($a < 18) {
        echo "Invalid age\n";
    }

```

```

else {
    echo "Enter weight in pounds: ";
    $w = trim(fgets(STDIN));
    echo "Enter height in inches: ";
    $h = trim(fgets(STDIN));

    $bmi = $w * 703 / $h ** 2;

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

```

22. Solution

```

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

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

if ($water < 0) {
    echo "Error! You entered a negative value\n";
}
else {
    if ($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, "\n";
}
?>

```

23. 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, "\n";
?>

```

24. Solution

```

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

if ($wind < 0) {
    echo "Error! You entered a negative value\n";
}
else {
    if ($wind < 1) {
        echo "Beaufort: 0\nCalm\n";
    }
    elseif ($wind < 4) {
        echo "Beaufort: 1\nLight air\n";
    }
}

```

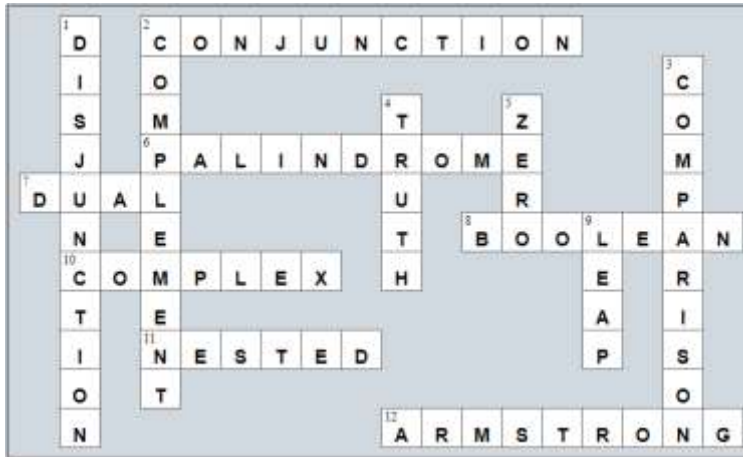
```
}
elseif ($wind < 8) {
    echo "Beaufort: 2\nLight breeze\n";
}
elseif ($wind < 13) {
    echo "Beaufort: 3\nGentle breeze\n";
}
elseif ($wind < 18) {
    echo "Beaufort: 4\nModerate breeze\n";
}
elseif ($wind < 25) {
    echo "Beaufort: 5\nFresh breeze\n";
}
elseif ($wind < 31) {
    echo "Beaufort: 6\nStrong breeze\n";
}
elseif ($wind < 39) {
    echo "Beaufort: 7\nModerate gale\n";
}
elseif ($wind < 47) {
    echo "Beaufort: 8\nGale\n";
}
elseif ($wind < 55) {
    echo "Beaufort: 9\nStrong gale\n";
}
elseif ($wind < 64) {
    echo "Beaufort: 10\nStorm\n";
}
elseif ($wind < 74) {
    echo "Beaufort: 11\nViolent storm\n";
}
else {
    echo "Beaufort: 12\nHurricane force\n";
}

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

Review in “Decision Control Structures”

Review Crossword Puzzle

1.



24.3 Review Questions: True/False

1. true

2. true

3. false

4. false

5. true

Chapter 24

25.4 Review Questions: True/False

- Chapter 25

1. true

2. false

3. false

4. false

5. false

6. false

7. true

8. true
9. false

10. false

11. false

12. true

13. false

14. false

15. true

16. false

25.5 Review Questions: Multiple Choice

1. c

2. c

3. a

4. b

5. d

6. b
7. c

8. b

9. b

10. d

11. a

12. d

25.6 Review Exercises

1. Solution

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

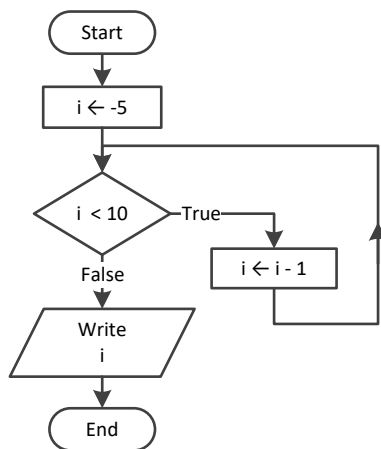
2. Solution

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

10	<code>\$i--</code>	0	3
11	<code>\$x += \$i</code>	0	3
12	<code>while (\$i >= 0)</code>	true	
13	<code>\$i--</code>	-1	3
14	<code>\$x += \$i</code>	-1	2
15	<code>while (\$i >= 0)</code>	false	
16	<code>echo \$x</code>	It displays: 2	

It performs 4 iterations

3. Solution



Step	Statement	Notes	\$i
1	<code>\$i = -5</code>		-5
2	<code>while (\$i < 10)</code>	true	
3	<code>\$i--</code>		-6
4	<code>while (\$i < 10)</code>	true	
5	<code>\$i--</code>		-7
6	<code>while (\$i < 10)</code>	true	
7	<code>\$i--</code>		-8
8
9

It performs an infinite number of iterations

4. Solution

Step	Statement	\$a	\$b	\$c	\$d
1	<code>\$a = 2</code>	2	?	?	?
2	<code>while (\$a <= 10)</code>	true			
3	<code>\$b = \$a + 1</code>	2	3	?	?
4	<code>\$c = \$b * 2</code>	2	3	6	?

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

5. Solution

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

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

6. Solution

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

7. Solution

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

8. Solution

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

9. Solution

Step	Statement	\$x	\$y
1	\$y = 2	?	2
2	\$x = 0	0	2
3	\$y = \$y ** 2	0	4
4	if (\$x < 256)	true	
5	\$x = \$x + \$y	4	
6	echo \$x, ", ", \$y	It displays: 4, 4	
7	while (\$y < 65535)	true	
8	\$y = \$y ** 2	4	16
9	if (\$x < 256)	true	
10	\$x = \$x + \$y	20	16

11	echo \$x, ", ", \$y	It displays: 20, 16	
12	while (\$y < 65535)	true	
13	\$y = \$y ** 2	20	256
14	if (\$x < 256)	true	
15	\$x = \$x + \$y	276	256
16	echo \$x, ", ", \$y	It displays: 276, 256	
17	while (\$y < 65535)	true	
18	\$y = \$y ** 2	276	65536
19	if (\$x < 256)	false	
20	echo \$x, ", ", \$y	It displays: 276, 65536	
21	while (\$y < 65535)	false	

10. Solution

Step	Statement	\$a	\$b	\$c	\$d	\$x
1	\$a = 2	2	?	?	?	?
2	\$b = 4	2	4	?	?	?
3	\$c = 0	2	4	0	?	?
4	\$d = 0	2	4	0	0	?
5	\$x = \$a + \$b	2	4	0	0	6
6	if (\$x % 2 != 0)	false				
7	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	while (\$c < 11)	true				
25	\$x = \$a + \$b	5	5	8	5	10
26	if (\$x % 2 != 0)	false				
27	elseif (\$d % 2 == 0)	false				
28	\$c = \$c + 3	5	5	11	5	10
29	\$a = \$b	5	5	11	5	10
30	\$b = \$d	5	5	11	5	10
31	while (\$c < 11)	false				

11. Solution

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

12. Solution

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

13. Solution

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

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

    echo $total, "\n";
    if ($n > 0) {
```

```
        echo $total / $n, "\n";
    }
?>
```

14. Solution

```
<?php
    $count = 0;

    $n = trim(fgets(STDIN));
    $p = 1;

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

    if ($count > 0) {
        echo $p, "\n";
    }
    else {
        echo "You entered no even integers\n";
    }
?>
```

15. Solution

```
<?php
    $total = 0;

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

16. Solution

```
<?php
    $total = 0;

    $i = 1;
    while ($i <= 20) {
```

```

    $a = trim(fgets(STDIN));
    if ($a >= 100 && $a <= 999) {
        $total = $total + $a;
    }
    $i++;
}
echo $total, "\n";
?>

```

17. Solution

```

<?php
    $p = 1;

    $a = trim(fgets(STDIN));
    while ($a != 0) {
        $p = $p * $a;
        $a = trim(fgets(STDIN));
    }
    echo $p, "\n";
?>

```

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, "\n"	It displays: 54	

18. Solution

```

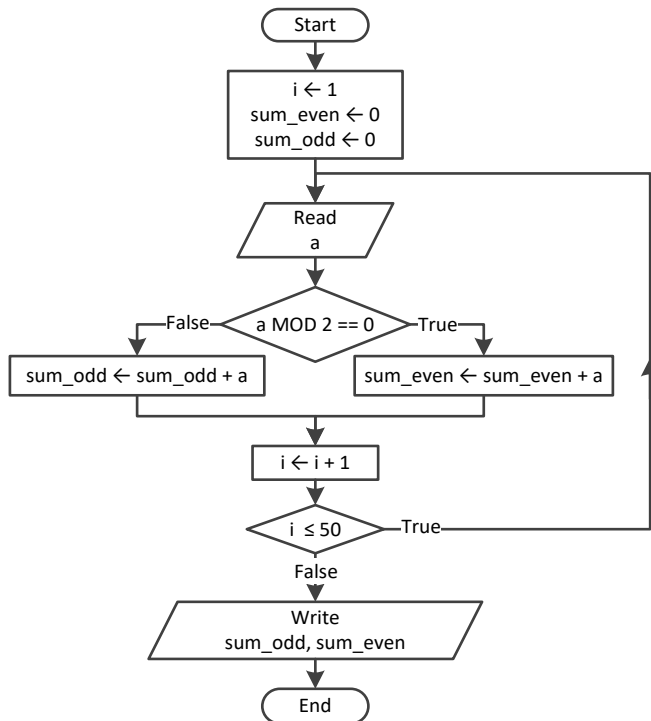
<?php
    $population = 30000;

    $years = 0;
    while ($population <= 100000) {
        $population += $population * 0.03;
        $years++;
    }
    echo $years, "\n";

```


?>

19. 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, "\n";
?>

```

20. 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), "\n";
?>
```

21. 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, "\n";
?>
```

22. Solution

```
<?php
$population = 50000;

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

Chapter 26

26.3 Review Questions: True/False

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

26.4 Review Questions: Multiple Choice

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

26.5 Review Exercises**1. Solution**

Step	Statement	\$a	\$b	\$j
1	\$a = 0	0	?	?
2	\$b = 0	0	0	?
3	\$j = 0	0	0	0
4	\$j <= 8	true		
5	if (\$j < 5)	true		
6	\$b++	0	1	0
7	\$j += 2	0	1	2
8	\$j <= 8	true		
9	if (\$j < 5)	true		
10	\$b++	0	2	2
11	\$j += 2	0	2	4
12	\$j <= 8	true		
13	if (\$j < 5)	true		
14	\$b++	0	3	4
15	\$j += 2	0	3	6
16	\$j <= 8	true		
17	if (\$j < 5)	false		
18	\$a += \$j - 1	5	3	6

19	<code>\$j += 2</code>	5	3	8
20	<code>\$j <= 8</code>	true		
21	<code>if (\$j < 5)</code>	false		
22	<code>\$a += \$j - 1</code>	12	3	8
23	<code>\$j += 2</code>	12	3	10
24	<code>\$j <= 8</code>	false		
25	<code>echo \$a, ", ", \$b</code>	It displays: 12, 3		

2. Solution

For input value of 10

Step	Statement	\$a	\$b	\$j
1	<code>\$a = trim(fgets(STDIN))</code>	10	?	?
2	<code>\$b = \$a</code>	10	10	?
3	<code>\$j = \$a - 5</code>	10	10	5
4	<code>\$j <= \$a</code>	true		
5	<code>if (\$j % 2 != 0)</code>	true		
6	<code>\$b = \$a + \$j + 5</code>	10	20	5
7	<code>\$j += 2</code>	10	20	7
8	<code>\$j <= a</code>	true		
9	<code>if (\$j % 2 != 0)</code>	true		
10	<code>\$b = \$a + \$j + 5</code>	10	22	7
11	<code>\$j += 2</code>	10	22	9
12	<code>\$j <= a</code>	true		
13	<code>if (\$j % 2 != 0)</code>	true		
14	<code>\$b = \$a + \$j + 5</code>	10	24	9
15	<code>\$j += 2</code>	10	24	11
16	<code>\$j <= \$a</code>	false		
17	<code>echo \$b</code>	It displays: 24		

For input value of 21

Step	Statement	\$a	\$b	\$j
1	<code>\$a = trim(fgets(STDIN))</code>	21	?	?
2	<code>\$b = \$a</code>	21	21	?
3	<code>\$j = \$a - 5</code>	21	21	16
4	<code>\$j <= \$a</code>	true		
5	<code>if (\$j % 2 != 0)</code>	false		
6	<code>\$b = \$a + \$j + 5</code>	21	5	16
7	<code>\$j += 2</code>	21	5	18

8	<code>\$j <= \$a</code>	true		
9	<code>if (\$j % 2 != 0)</code>	false		
10	<code>\$b = \$a + \$j + 5</code>	21	3	18
11	<code>\$j += 2</code>	21	3	20
12	<code>\$j <= \$a</code>	true		
13	<code>if (\$j % 2 != 0)</code>	false		
14	<code>\$b = \$a + \$j + 5</code>	21	1	20
15	<code>\$j += 2</code>	21	1	22
16	<code>\$j <= \$a</code>	false		
17	<code>echo \$b</code>	It displays: 1		

3. Solution

For input value of 12

Step	Statement	\$a	\$x	\$y	\$j
1	<code>\$a = trim(fgets(STDIN))</code>	12	?	?	?
2	<code>\$j = 2</code>	12	?	?	2
3	<code>\$j <= \$a - 1</code>	true			
4	<code>\$x = \$j * 3 + 3</code>	12	9	?	2
5	<code>\$y = \$j * 2 + 10</code>	12	9	14	2
6	<code>if (\$y - \$x > 0 \$x > 30)</code>	true			
7	<code>\$y *= 2</code>	12	9	28	2
8	<code>\$x += 4</code>	12	13	28	2
9	<code>echo \$x, ", ", \$y</code>	It displays: 13, 28			
10	<code>\$j += 3</code>	12	13	28	5
11	<code>\$j <= \$a - 1</code>	true			
12	<code>\$x = \$j * 3 + 3</code>	12	18	28	5
13	<code>\$y = \$j * 2 + 10</code>	12	18	20	5
14	<code>if (\$y - \$x > 0 \$x > 30)</code>	true			
15	<code>\$y *= 2</code>	12	18	40	5
16	<code>\$x += 4</code>	12	22	40	5
17	<code>echo \$x, ", ", \$y</code>	It displays: 22, 40			
18	<code>\$j += 3</code>	12	22	40	8
19	<code>\$j <= \$a - 1</code>	true			
20	<code>\$x = \$j * 3 + 3</code>	12	27	40	8
21	<code>\$y = \$j * 2 + 10</code>	12	27	26	8
22	<code>if (\$y - \$x > 0 \$x > 30)</code>	false			
23	<code>\$x += 4</code>	12	31	26	8

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

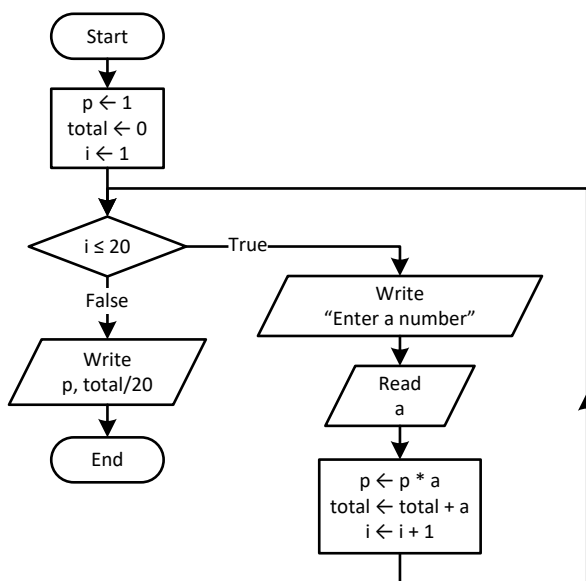
4. Solution

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

5. Solution

It displays: sueZ

6. Solution



```

<?php
    $p = 1;
    $total = 0;
    for ($i = 1; $i <= 20; $i++) {

```

```

    echo "Enter a number: ";
    $a = trim(fgets(STDIN));
    $p = $p * $a;
    $total = $total + $a;
}
echo $p, "\n";
echo $total / 20, "\n";
?>

```

7. Solution

```

<?php
for ($i = 0 ; $i <= 360; $i += 0.5) {
    echo sin($i * pi() / 180), "\n";
}
?>

```

8. Solution

```

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

```

9. Solution

```

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

```

10. Solution

```

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

```

11. Solution

```

<?php
$s = 0;

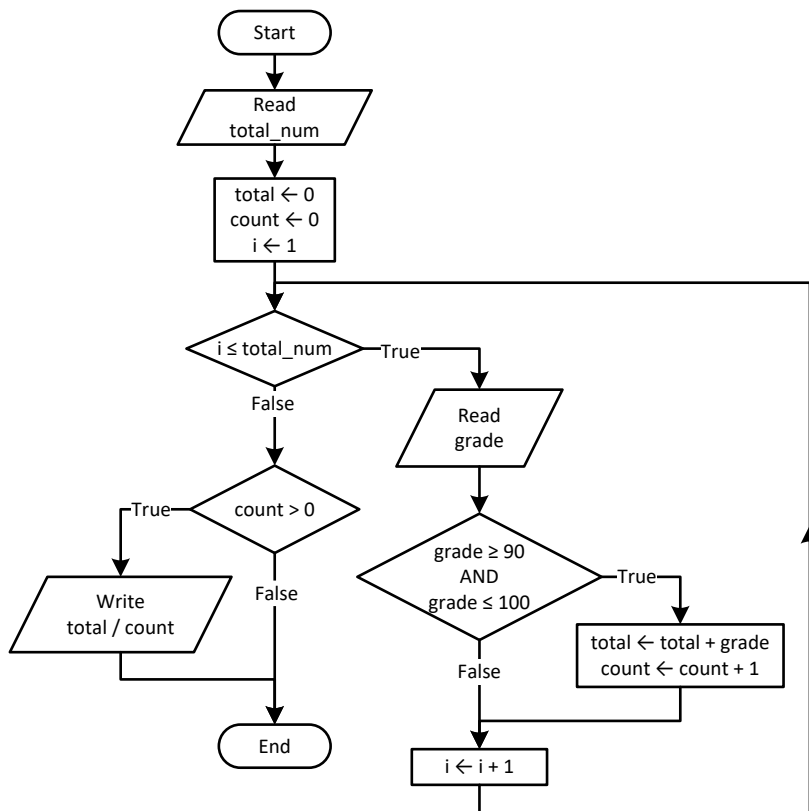
```

```

$i = 1;
$offset = 0;
while ($i <= 191) {
    $s += $i;
    $offset++;
    $i += $offset;
}
echo $s, "\n";
?>

```

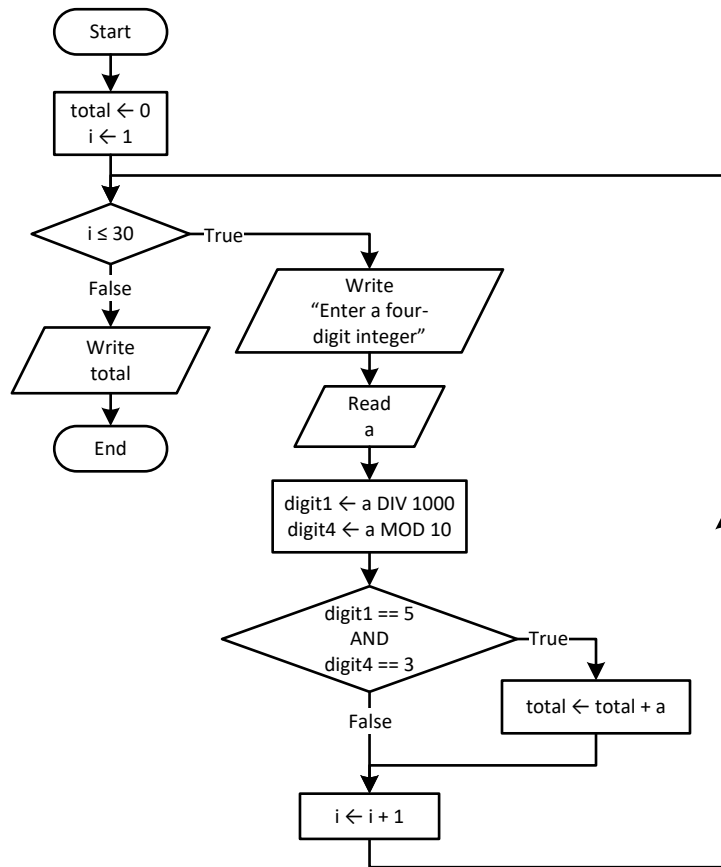
12. Solution



```

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

```

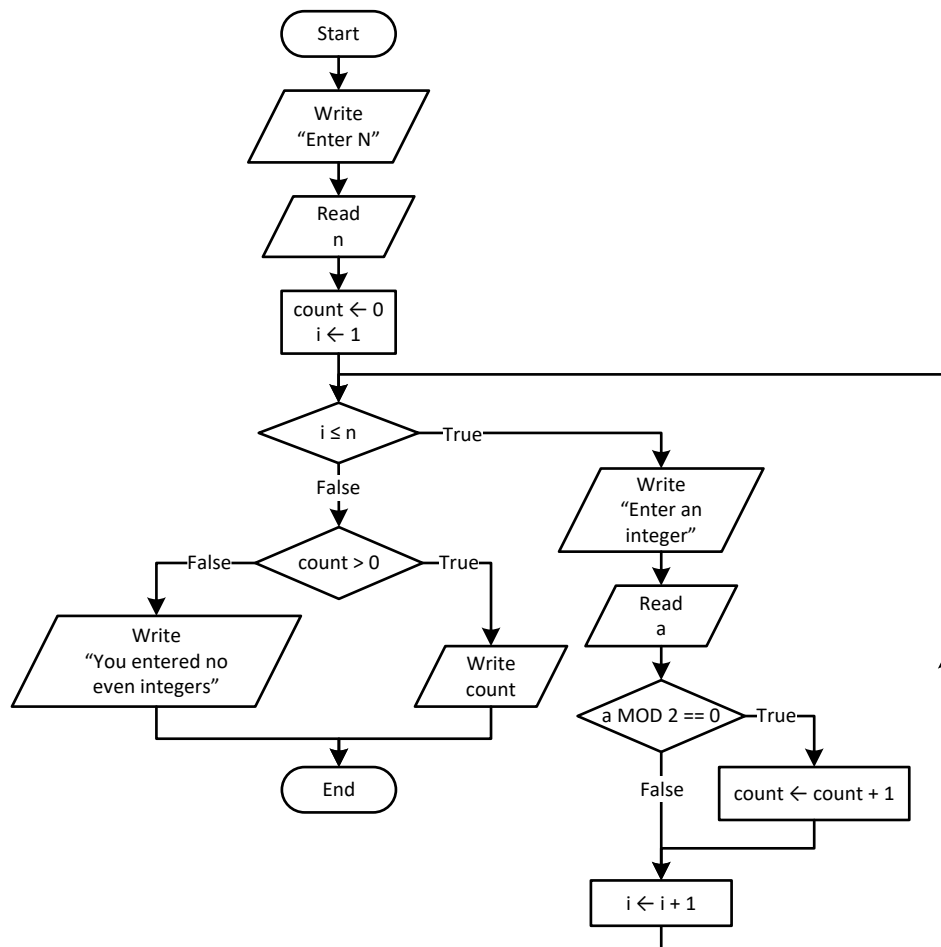

13. Solution

```

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

```

14. Solution

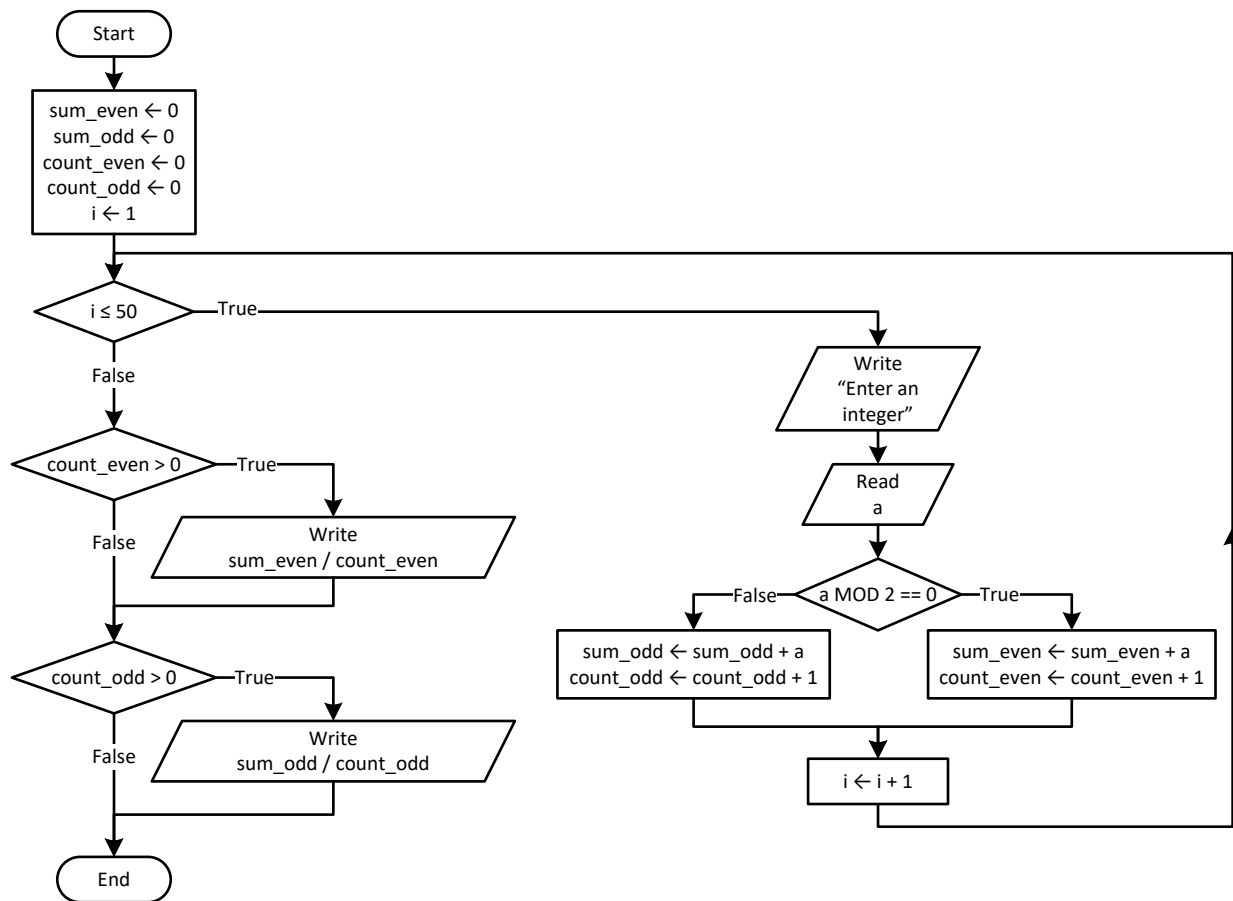


```

<?php
echo "Enter N: ";
$n = trim(fgets(STDIN));
$count = 0;
for ($i = 1; $i <= $n; $i++) {
    echo "Enter an integer: ";
    $a = trim(fgets(STDIN));
    if ($a % 2 == 0) {
        $count++;
    }
}
if ($count > 0) {
    echo $count, "\n";
}
else {
    echo "You entered no even integers\n";
}
?>

```

15. Solution

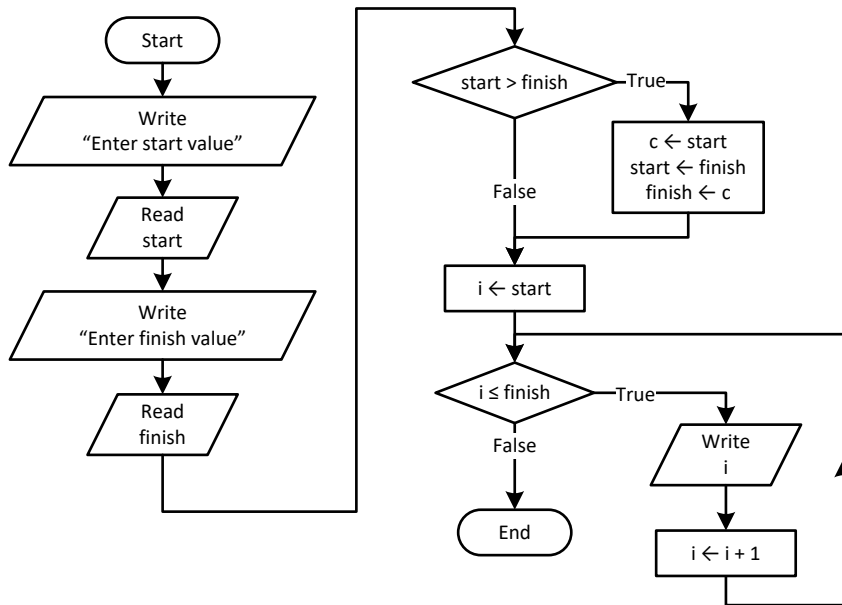


```

<?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 / $count_even, "\n";
}
if ($count_odd > 0) {
    echo $sum_odd / $count_odd, "\n";
}

```

| ?>

16. Solution

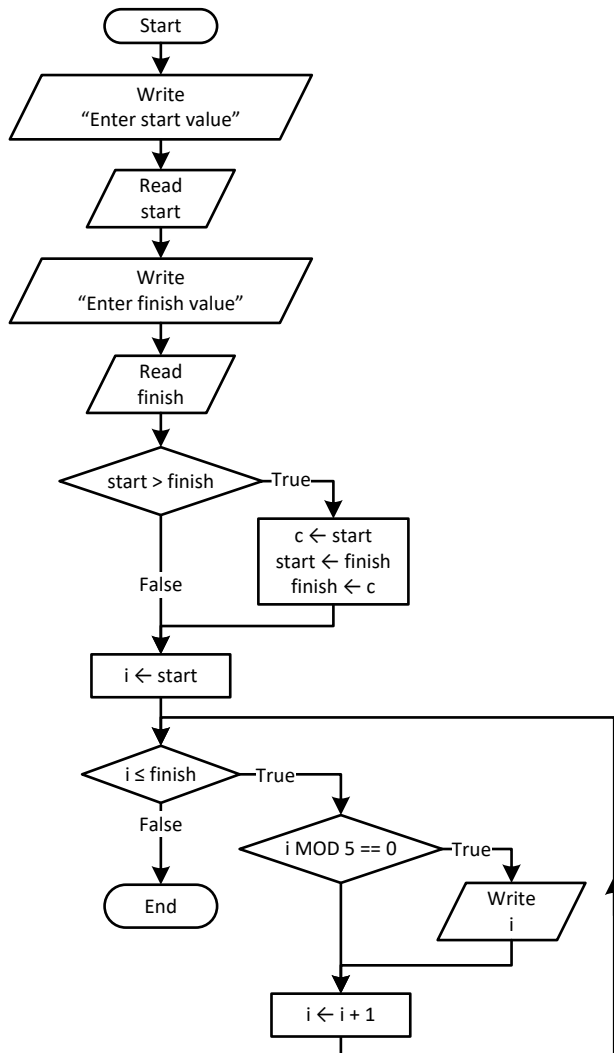
```

<?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, "\n";
}
?>

```

17. 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, "\n";
    }
}
?>

```

18. Solution

First approach

```
<?php
    echo "Enter a value for base: ";
    $b = 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 *= $b;
        }
    }
    else {
        for ($i = 1; $i <= -$exp; $i++) {
            $p *= 1 / $b;
        }
    }
    echo $p, "\n";
?>
```

Second approach

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

    $p = 1;
    for ($i = 1; $i <= abs($exp); $i++) {
        $p *= $b;
    }
    if ($exp < 0) {
        $p = 1 / $p;
    }
    echo $p, "\n";
?>
```

19. Solution

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

    $count = 0;
    for ($i = 0; $i <= strlen($msg) - 1; $i++) {
        $character = $msg[$i];
        if ($character == " ") {
            $count++;
        }
    }
```

```

    }
    $words = $count + 1;

    echo "The message entered contains ", $words, " words\n";
?>

```

20. Solution

```

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

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

    $words = $count + 1;
    echo "The average number of letters in each word is ";
    echo ($characters - $count) / $words, "\n";
?>

```

21. Solution

```

<?php
    $consonants = "BCDFGHJKLMNPQRSTVWXYZ";

    echo "Enter an English message: ";
    $message = strtoupper(trim(fgets(STDIN)));

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

        if (strpos($consonants, $character) !== false) { //If character is found in consonants
            $count++;
        }
    }
    echo "Consonants: ", $count, "\n";
?>

```

22. Solution

```

<?php
    $vowels = "AEIOU";
    $consonants = "BCDFGHJKLMNPQRSTVWXYZ";
    $digits = "0123456789";

    echo "Enter an English message: ";

```

```
$message = strtoupper(trim(fgets(STDIN)));

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

    if (strpos($vowels, $character) != false) { //If character is found in vowels
        $countv++;
    }
    elseif (strpos($consonants, $character) != false) { //If character is found in consonants
        $countc++;
    }
    elseif (strpos($digits, $character) != false) { //If character is found in digits
        $countd++;
    }
}
echo "Vowels: ", $countv, "\n";
echo "Consonants: ", $countc, "\n";
echo "Digits: ", $countd, "\n";
?>
```


27.3 Review Questions: True/False

1. true

2. true

3. false

4. true

5. true
6. false

7. true

8. true

9. true

10. true

27.4 Review Questions: Multiple Choice

1. b

2. a

3. c
4. a

5. b

27.5 Review Exercises

1. Solution

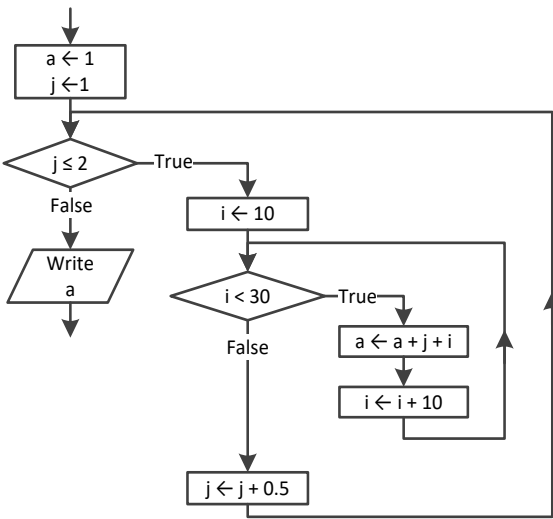
- i. 10

ii. A value greater than or equal to 4.5 and less than 5.0

iii. -7 (or -6)

iv. 138 (or 139)

2. Solution



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

6	$\$a = \$a + \$j + \i	12	10	1
7	$\$i += 10$	12	20	1
8	$\$i < 30$	true		
9	$\$a = \$a * \$j + \i	33	20	1
10	$\$i += 10$	33	30	1
11	$\$i < 30$	false		
12	$\$j += 0.5$	33	30	1.5
13	$\$j \leq 2$	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	$\$j \leq 2$	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	$\$j \leq 2$	false		
34	echo \$a	It displays: 100		

3. Solution

Step	Statement	\$s	\$i	\$j
1	$\$s = 0$	0	?	?
2	$\$i = 1$	0	1	?
3	$\$i \leq 4$	true		
4	$\$j = 3$	0	1	3
5	$\$j \geq \i	true		

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

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

4. Solution

For input value of “NO”

Step	Statement	\$s	\$y	\$i	\$ans
1	\$s = 1	1	?	?	?
2	\$y = 25	1	25	?	?
3	\$i = 1	1	25	1	?
4	\$i <= 3	true			
5	\$s = \$s + \$y	26	25	1	?
6	\$y -= 5	26	20	1	?
7	\$i++	26	20	2	?
8	\$i <= 3	true			
9	\$s = \$s + \$y	46	20	2	?
10	\$y -= 5	46	15	2	?
11	\$i++	46	15	3	?
12	\$i <= 3	true			
13	\$s = \$s + \$y	61	15	3	?
14	\$y -= 5	61	10	3	?
15	\$i++	61	10	4	?
16	\$i <= 3	false			
17	\$ans = trim(fgets(STDIN))	61	10	4	"NO"
18	while (\$ans == "YES")	false			
19	echo \$s	It displays: 61			

For input values of "YES", "NO"

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

16	<code>\$i <= 3</code>	false			
17	<code>\$ans = trim(fgets(STDIN))</code>	61	10	4	"YES"
18	<code>while (\$ans == "YES")</code>	true			
19	<code>\$i = 1</code>	61	10	1	"YES"
20	<code>\$i <= 3</code>	true			
21	<code>\$s = \$s + \$y</code>	71	10	1	"YES"
22	<code>\$y -= 5</code>	71	5	1	"YES"
23	<code>\$i++</code>	71	5	2	"YES"
24	<code>\$i <= 3</code>	true			
25	<code>\$s = \$s + \$y</code>	76	5	2	"YES"
26	<code>\$y -= 5</code>	76	0	2	"YES"
27	<code>\$i++</code>	76	0	3	"YES"
28	<code>\$i <= 3</code>	true			
29	<code>\$s = \$s + \$y</code>	76	0	3	"YES"
30	<code>\$y -= 5</code>	76	-5	3	"YES"
31	<code>\$i++</code>	76	-5	4	"YES"
32	<code>\$i <= 3</code>	false			
33	<code>\$ans = trim(fgets(STDIN))</code>	76	-5	4	"NO"
34	<code>while (\$ans == "YES")</code>	false			
35	<code>echo \$s</code>	It displays: 76			

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

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

16	<code>\$i <= 3</code>	false			
17	<code>\$ans = trim(fgets(STDIN))</code>	61	10	4	"YES"
18	<code>while (\$ans == "YES")</code>	true			
19	<code>\$i = 1</code>	61	10	1	"YES"
20	<code>\$i <= 3</code>	true			
21	<code>\$s = \$s + \$y</code>	71	10	1	"YES"
22	<code>\$y -= 5</code>	71	5	1	"YES"
23	<code>\$i++</code>	71	5	2	"YES"
24	<code>\$i <= 3</code>	true			
25	<code>\$s = \$s + \$y</code>	76	5	2	"YES"
26	<code>\$y -= 5</code>	76	0	2	"YES"
27	<code>\$i++</code>	76	0	3	"YES"
28	<code>\$i <= 3</code>	true			
29	<code>\$s = \$s + \$y</code>	76	0	3	"YES"
30	<code>\$y -= 5</code>	76	-5	3	"YES"
31	<code>\$i++</code>	76	-5	4	"YES"
32	<code>\$i <= 3</code>	false			
33	<code>\$ans = trim(fgets(STDIN))</code>	76	-5	4	"YES"
34	<code>while (\$ans == "YES")</code>	true			
35	<code>\$i = 1</code>	76	-5	1	"YES"
36	<code>\$i <= 3</code>	true			
37	<code>\$s = \$s + \$y</code>	71	-5	1	"YES"
38	<code>\$y -= 5</code>	71	-10	1	"YES"
39	<code>\$i++</code>	71	-10	2	"YES"
40	<code>\$i <= 3</code>	true			
41	<code>\$s = \$s + \$y</code>	61	-10	2	"YES"
42	<code>\$y -= 5</code>	61	-15	2	"YES"
43	<code>\$i++</code>	61	-15	3	"YES"
44	<code>\$i <= 3</code>	true			
45	<code>\$s = \$s + \$y</code>	46	-15	3	"YES"
46	<code>\$y -= 5</code>	46	-20	3	"YES"
47	<code>\$i++</code>	46	-20	4	"YES"
48	<code>\$i <= 3</code>	false			
49	<code>\$ans = trim(fgets(STDIN))</code>	46	-20	4	"NO"
50	<code>while (\$ans == "YES")</code>	false			
51	<code>echo \$s</code>	It displays: 46			

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: ";
    $y = trim(fgets(STDIN));

    for ($i = 1; $i <= $y; $i++) {
        for ($j = 1; $j <= $y; $j++) {
            echo "* ";
        }
    }
}
```

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

10. Solution

```
<?php  
echo "Enter an integer between 3 and 20: ";  
$y = trim(fgets(STDIN));  
  
for ($j = 1; $j <= $y; $j++) {  
    echo "* ";  
}  
echo "\n";  
  
for ($i = 1; $i <= $y - 2; $i++) {  
    echo "* ";  
    for ($j = 1; $j <= $y - 2; $j++) {  
        echo " ";  
    }  
    echo "* \n";  
}  
  
for ($j = 1; $j <= $y; $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 28

28.8 Review Questions: True/False

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

28.9 Review Questions: Multiple Choice

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

28.10 Review Exercises**1. Solution**

```

$count_names = 0;
$count_not_johns = 0;
$name = "";
echo "Enter a name: ";
$name = trim(fgets(STDIN));
while ($name != "STOP") {
    echo "Enter a name: ";
    $name = trim(fgets(STDIN));
    $count_names++;
    if ($name != "John") {
        $count_not_johns++;
    }
    echo "Enter a name: ";
    $name = Console.ReadLine();
}
echo $count_names, " names entered\n";
echo "Names other than John entered ", $count_not_johns, " times\n";

```

2. Solution**First approach**

```

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

$found = false;
for ($i = 0; $i <= strlen($text) - 1; $i++) {
    $character = $text[$i];
    if ($character == " ") {
        $found = true;
    }
}

```

```

        break;
    }
}

if (!$found) {
    echo "One Single Word\n";
}
else {
    echo "Complete Sentence\n";
}
?>

```

Second approach

```

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

if (strpos($text, " ") !== false) {
    echo "Complete Sentence\n";
}
else {
    echo "One Single Word\n";
}
?>

```

3. Solution

First approach

```

<?php
$digits = "0123456789";

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

$found = false;
for ($i = 0; $i <= strlen($sentence) - 1; $i++) {
    $character = $sentence[$i];
    if (strpos($digits, $character) !== false) {
        $found = true;
        break;
    }
}

if ($found) {
    echo "The sentence contains a number\n";
}
?>

```

Second approach

```

<?php
echo "Enter a text: ";
$sentence = trim(fgets(STDIN));

$found = false;
for ($i = 0; $i <= 9; $i++) {

```

```

    $digit = $i;
    if (strpos($sentence, $digit) !== false) {
        $found = true;
        break;
    }
}

if ($found) {
    echo "The sentence contains a number\n";
}
?>

```

4. Solution

```

echo "Printing all integers from 1 to 100\n";
$i = 1;
while ($i < 101) {
    echo $i;
    $i++;
}

```

5. Solution

```

echo "Printing odd integers from 1 to 99\n";
$i = 1;
while ($i < 100) {
    echo $i;
    $i += 2;
}

```

6. Solution

```

$s = 0;
for ($i = 1; $i <= 100; $i++) {
    $number = trim(fgets(STDIN));
    $s = $s + $number;
}
$average = $s / 100.0;
echo $average;

```

7. Solution

```

$s = 0;

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

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

```

8. Solution

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

9. 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";
    }
?>
```

10. Solution

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

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

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

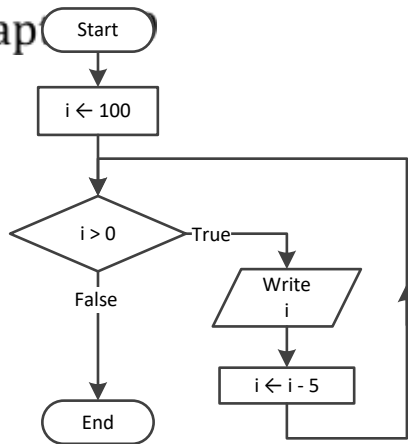
    for ($i = 1; $i <= $n; $i++) {
        echo $i, "\t|\t";
    }
}
```

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

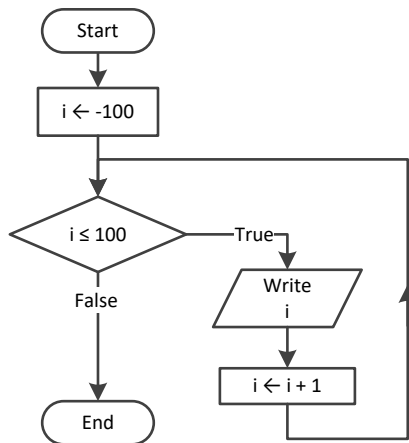
29.4 Review Exercises

1. Solution

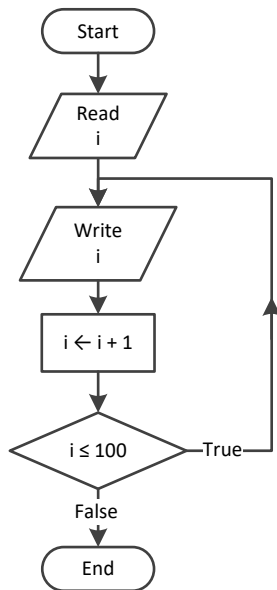
Chap



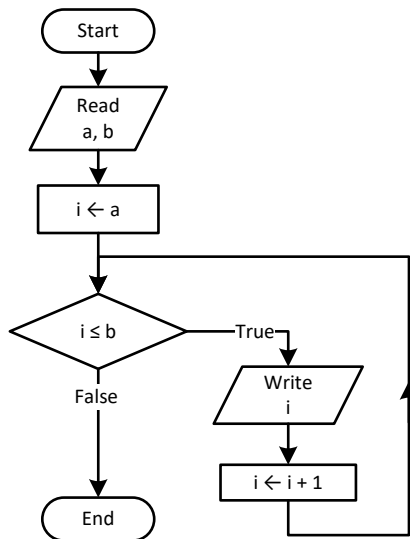
2. Solution



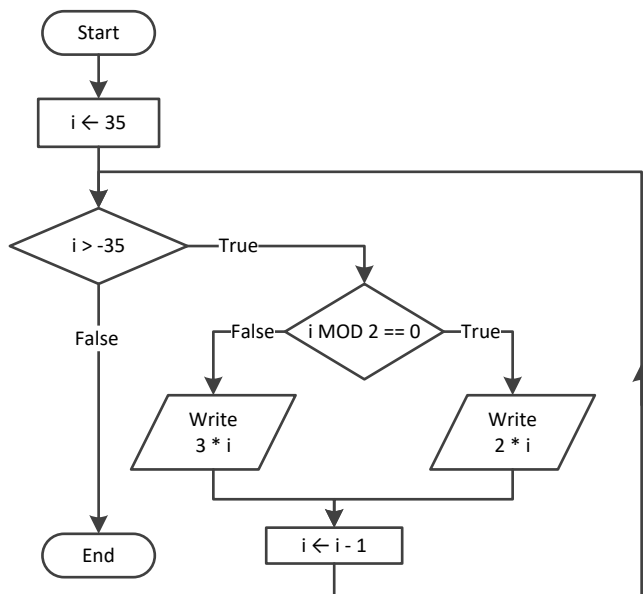
3. Solution



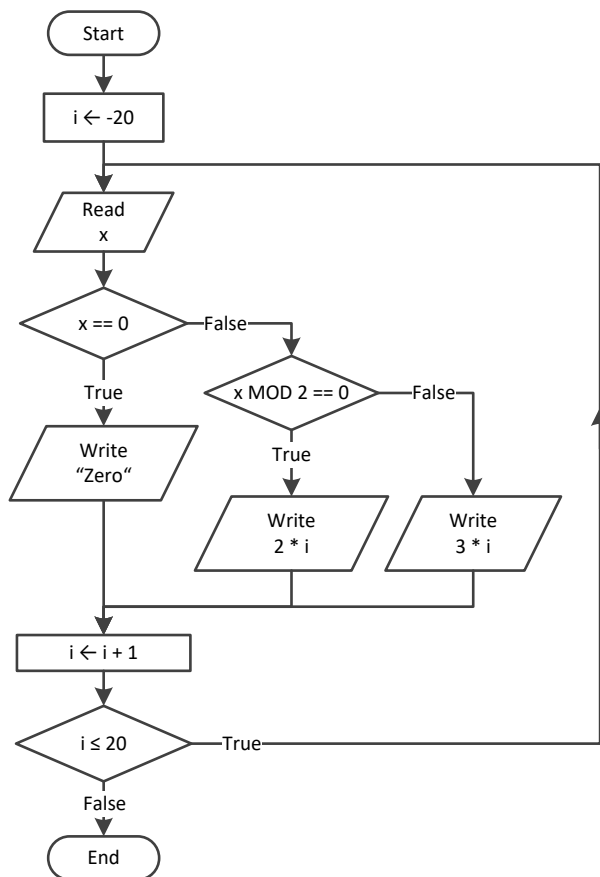
4. Solution



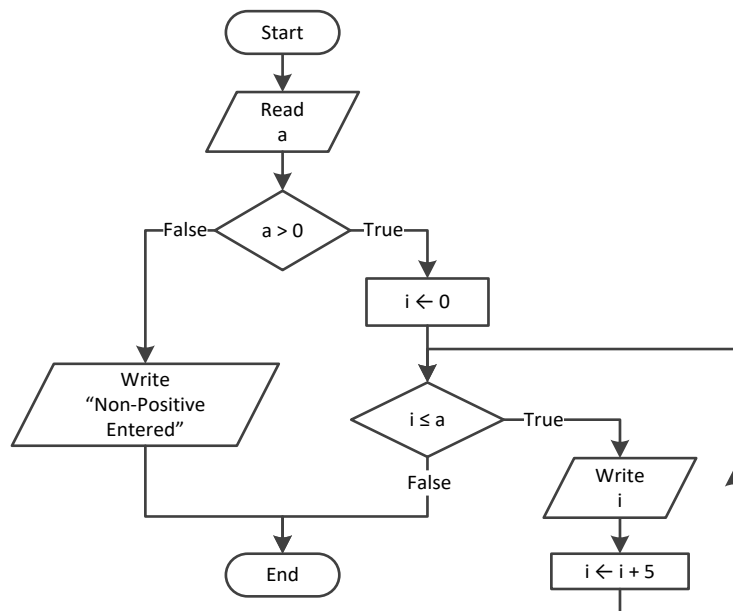
5. Solution



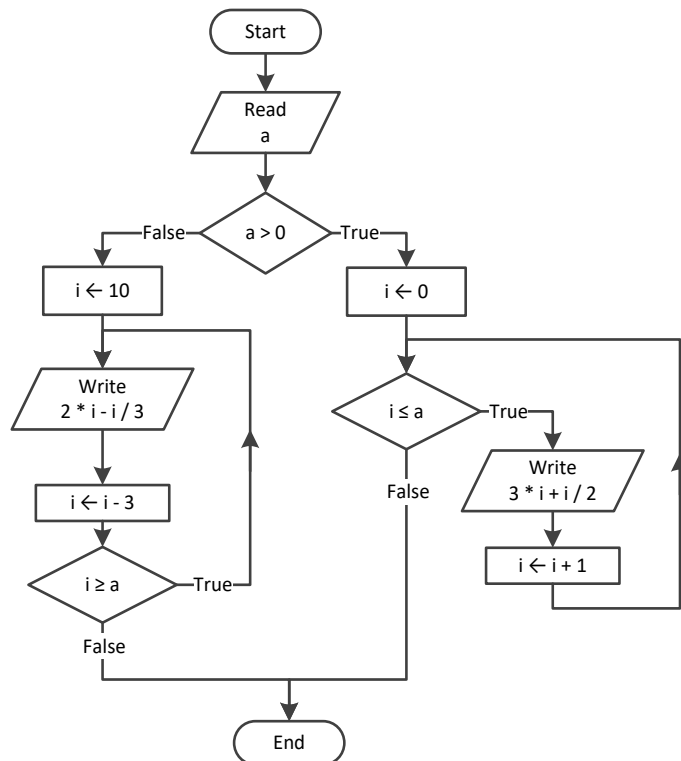
6. Solution

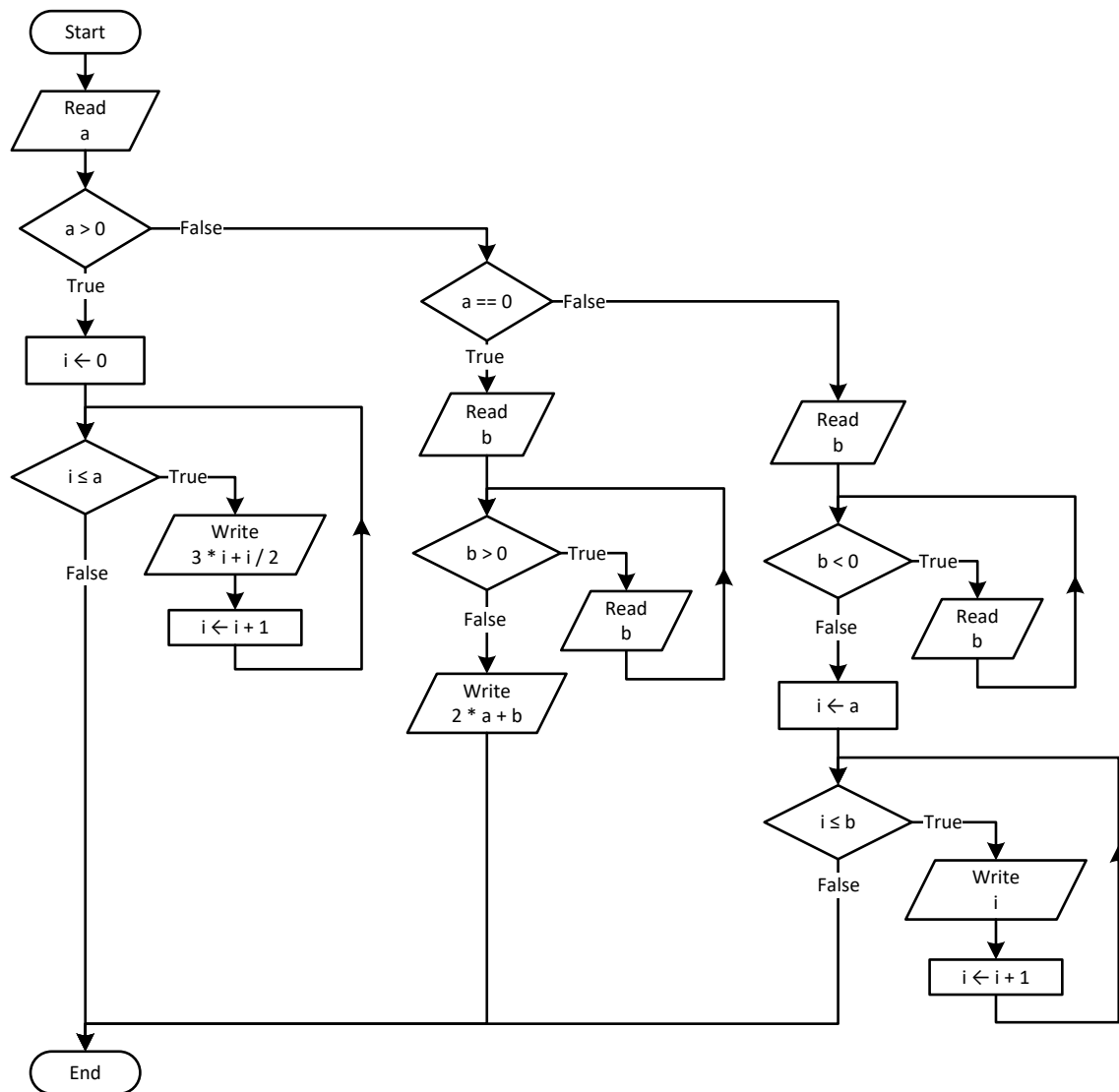


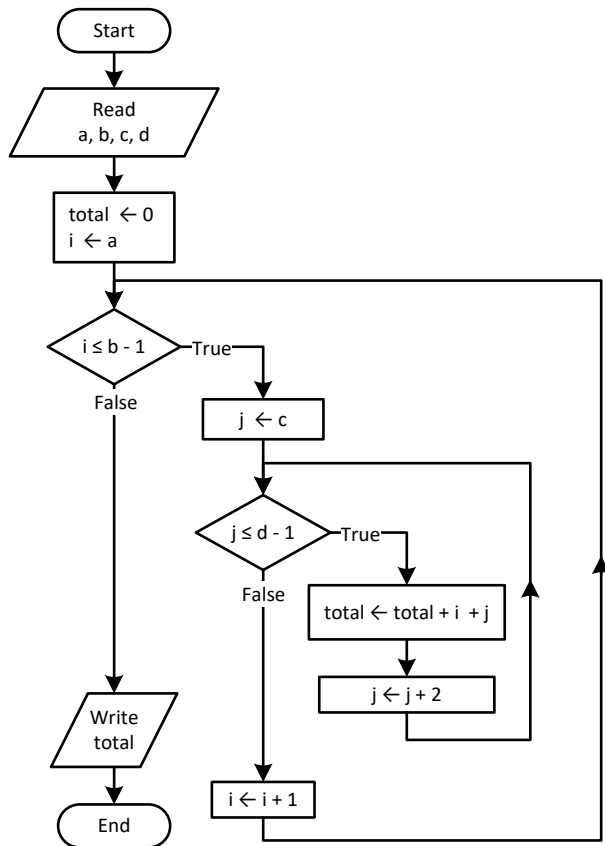
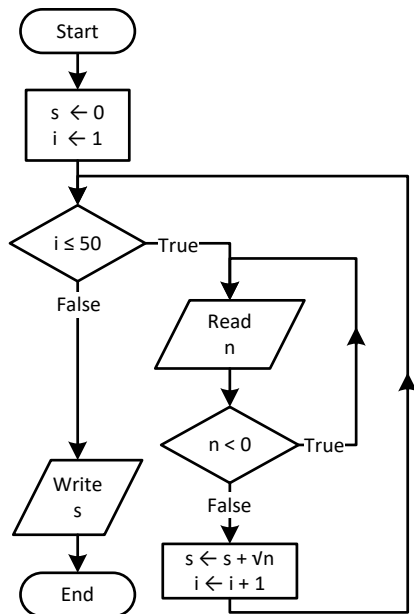
7. Solution

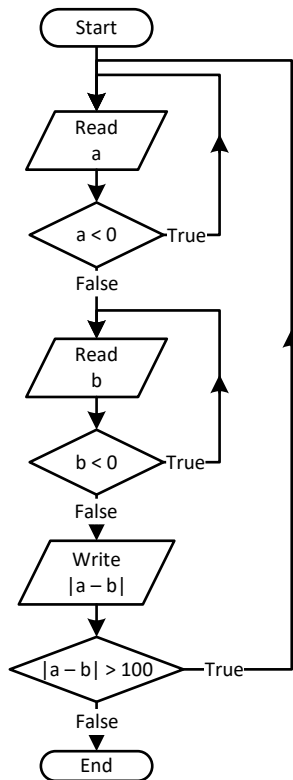
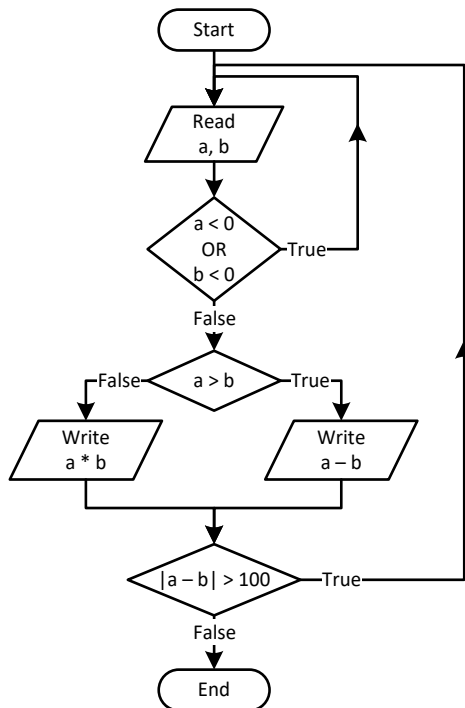


8. Solution



9. Solution

10. Solution**11. Solution**

12. Solution**13. Solution**

14. Solution

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

15. Solution

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

16. 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));
    }
?>
```

17. Solution

```
<?php
    $i = 1;
    $S = 0;
    $P = 1;
    $a = 0;

    while (true) {
        if ($i < 45) {
            $S += $a;
        }
        else {
            $P *= $a;
        }
    }
```

```
    $i++;  
    if ($i >= 90) break;  
    $a = trim(fgets(STDIN));  
}  
  
echo $S, " ", $P;  
?>
```

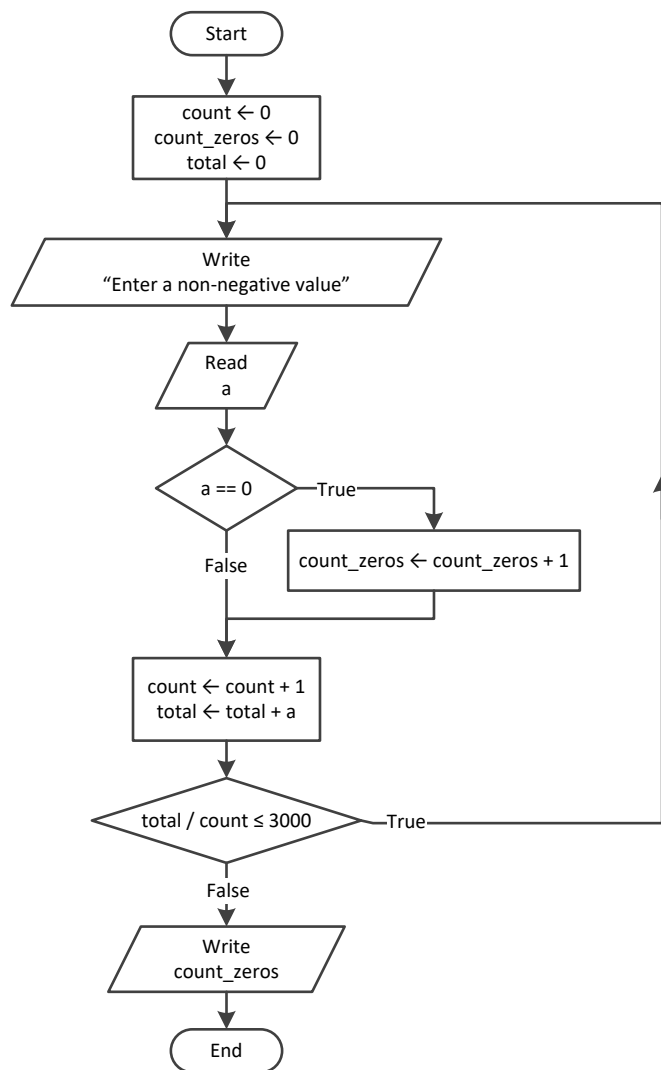
30.7 Review Questions: True/False

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

Chapter 30

30.8 Review Exercises

1. Solution



```

<?php
$count = 0;
$count_zeros = 0;
$total = 0;
do {
    echo "Enter a non-negative value: ";

```

```

    $a = trim(fgets(STDIN));
    if ($a == 0) {
        $count_zeros++;
    }
    $count++;
    $total += $a;
} while ($total / $count <= 3000);
echo $count_zeros, "\n";
?>

```

2. 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;
    $r = (int)($i / 10);
    $d3 = $r % 10;
    $r = (int)($r / 10);
    $d2 = $r % 10;
    $d1 = (int)($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";
                }
            }
        }
    }
}
?>

```

3. Solution

First approach

```

<?php
for ($i = 1000; $i <= 9999; $i++) {
    $d4 = $i % 10;

```



```

    $r = (int)($i / 10);
    $d3 = $r % 10;
    $r = (int)($r / 10);
    $d2 = $r % 10;
    $d1 = (int)($r / 10);
    if ($d1 > $d2 && $d2 == $d3 && $d3 < $d4) {
        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";
                }
            }
        }
    }
}
?>

```

4. Solution

First approach

```

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

$count = 0;

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

echo $count, "\n";
?>

```

Second approach

```

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

//Convert the absolute value of $x to string and get its length
$count = strlen((string)abs($x));
echo $count, "\n";
?>

```

5. Solution

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

6. Solution

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

7. Solution

```
<?php
    echo "Enter a non-negative number: ";
    $x = trim(fgets(STDIN));
    $count = 0;
    while ($x < 0) {
        $count++;
        if ($count == 2) break;

        echo "Error: Invalid number!\n";
        echo "Enter a non-negative number: ";
        $x = trim(fgets(STDIN));
    }

    if ($count < 2) {
        $y = sqrt($x);
        echo $y, "\n";
    }
    else {
        echo "Dude, you are dumb!\n";
    }
?>
```

8. 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() * $r ** 2;
        echo "The area is: ", $area, "\n";
    }
```

```

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

?>

```

9. Solution

```

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

?>

```

10. Solution

```

<?php
for ($x = -10; $x <= 10; $x++) {
    for ($y = -10; $y <= 10; $y++) {
        for ($z = -10; $z <= 10; $z++) {
            if (($x + $y) / 2.0 + 3.0 * $z ** 2 / ($x + 3 * $y + 45) == $x / 3.0) {
                echo $x, ", ", $y, ", ", $z, "\n";
            }
        }
    }
}

?>

```

11. 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;
    $m2 = (int)($m2 / 2);
}

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

$s = 0;
while ($m2 != 0) {

```

```

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

echo $s, "\n";
?>

```

12. Solution

```

<?php
$x = trim(fgets(STDIN));
while ($x <= 0) {
    echo "Error! You must enter a positive integer\n";
    $x = trim(fgets(STDIN));
}

$number_of_divisors = 2;
for ($i = 2; $i <= (int)($x / 2); $i++) {
    if ($x % $i == 0) {
        $number_of_divisors++;
    }
}
echo $number_of_divisors, "\n";
?>

```

13. Solution

```

<?php
echo "Enter an integer greater than 1: ";
$x = trim(fgets(STDIN));
while ($x <= 1) {
    echo "Error!\n";
    $x = trim(fgets(STDIN));
}

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

if ($number_of_divisors == 2) {
    echo "Number ", $x, " is prime\n";
}
?>

```

14. 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 <= (int)($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";
        }
    }
?>
```

15. Solution

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

    echo "Enter a second positive four-digit integer: ";
    $b = trim(fgets(STDIN));
    while ($b < 1000 || $b > 9999) {
        echo "Wrong number. Please enter a second positive 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;
    $r = (int)($x / 10);
    $d3 = $r % 10;
    $r = (int)($r / 10);
    $d2 = $r % 10;
    $d1 = (int)($r / 10);

    if ($d1 == $d4 && $d2 == $d3) {
        echo $x, "\n";
    }
}
?>

```

16. Solution

```

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

```

17. Solution

```

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

```

18. Solution

```

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

```

19. Solution

First approach

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

Second approach

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

20. 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;
}
?>
```

21. 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;
}
?>
```

22. Solution

```
<?php
    echo "Enter a positive integer: ";
    $n = trim(fgets(STDIN));
    while ($n <= 0) {
        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, "\n";
?>
```

23. Solution

```
<?php
    echo "Enter a positive integer: ";
    $n = trim(fgets(STDIN));
    while ($n <= 0) {
        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, "\n";
?>
```

24. Solution

```
<?php
    echo "Enter a positive integer: ";
    $n = trim(fgets(STDIN));
    while ($n <= 0) {
        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, "\n";
?>

```

25. Solution

```

<?php
echo "Enter a positive integer: ";
$n = trim(fgets(STDIN));
while ($n <= 0) {
    echo "Wrong number. Please enter a positive integer: ";
    $n = trim(fgets(STDIN));
}

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

echo $y, "\n";
?>

```

26. Solution


```

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

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

echo $factorial, "\n";
?>

```

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

27. 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 += $x ** $i / $factorial;

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

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

```

Second approach

```

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

$x = trim(fgets(STDIN));

$exponential = 1;
$i = 1;
$factorial = 1;
do {
    $exponential_previous = $exponential;

    $factorial *= $i;

    $exponential += $x ** $i / $factorial;

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

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

```

28. 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 * $x ** $i / $factorial;

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

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

```

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 * $x ** $i / $factorial;

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

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

```

29. 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 * $x ** $i / $factorial;

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

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

```

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 * $x ** $i / $factorial;

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

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

```

30. Solution

```

<?php
$maximum = -460;
$total = 0;
for ($i = 1; $i <= 31; $i++) {
    echo "Enter temperature for day ", $i, ": ";
    $t = trim(fgets(STDIN));
    while ($t < -459.67) {
        echo "Error! Wrong temperature.\n";
        echo "Enter temperature for day ", $i, ": ";
        $t = trim(fgets(STDIN));
    }

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

```

```

    }
}

echo $total / 31, " ", $maximum, "\n";
?>

```

31. Solution

```

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

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

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

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

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

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

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

    echo $maximum, " ", $max_hour, " ", $max_minutes, "\n";
    echo $minimum, " ", $min_hour, " ", $min_minutes, "\n";
}
?>

```

32. Solution

```

<?php
$alphabet = "abcdefghijklmnopqrstuvwxyz";

do {
    echo "Enter an integer between 1 and 26: ";
}

```

```

$a = trim(fgets(STDIN));

$failure = false;
if ($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);

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

    $failure = false;
    if ($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);

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

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

```

33. 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, "\n";
?>

```

34. Solution

```

<?php
$attempts = 0;
$first_player_attempts = 0;

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

```

35. Solution

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

36. Solution

```
<?php
echo "Enter total number of students: ";
$n = trim(fgets(STDIN));
while ($n <= 0) {
    echo "Wrong number. Please enter total number of students: ";
    $n = trim(fgets(STDIN));
}

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

$maximum = -1;
$minimum = 101;

for ($i = 1; $i <= $n; $i++) {
    echo "Enter grade for student No ", $i, ": ";
```



```
$grade = trim(fgets(STDIN));
while ($grade < 0 || $grade > 100) {
    echo "Wrong grade. Please enter grade for student No ", $i, ": ";
    $grade = trim(fgets(STDIN));
}

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

if ($grade >= 90 && $grade <= 100) {
    $total_a += $grade;
    $count_a++;
    if ($gender == "M") {
        $total_a_boys += $grade;
        $count_a_boys++;
    }
}
elseif ($grade >= 80 && $grade <= 89) {
    $total_b += $grade;
    $count_b++;
}
else {
    if ($gender == "F") {
        $count_cdef_girls++;
    }
}

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

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

$total += $grade;
}

if ($count_a > 0) {
    echo "The average value of those who got an 'A' is: ";
    echo $total_a / $count_a, "\n";
}
if ($count_b > 0) {
    echo "The average value of those who got a 'B' is: ";
    echo $total_b / $count_b, "\n";
}
if ($count_a_boys > 0) {
    echo "The average value of boys who got an 'A' is: ";
}
```

```

    echo $total_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: ", $maximum, "\n";
echo "The lowest grade is: ", $minimum, "\n";
echo "The average grade of the whole class is: ", $total / $n, "\n";
?>

```

37. Solution

```

<?php
do {
    echo "Enter amount: ";
    $amount = trim(fgets(STDIN));
    while ($amount <= 0) {
        echo "Wrong amount. Please enter amount: \n";
        $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? ";
    $answer = strtoupper(trim(fgets(STDIN)));
} while ($answer == "YES");
?>

```

38. Solution

```

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

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

Review in “Loop Control Structures”

Review Crossword Puzzle

1.



31.13 Review Questions: True/False

- Chapter 31
1. true

2. true

3. false

4. false

5. false

6. true

7. false

8. true

9. false

10. true

11. true

12. true

13. false

14. false

15. false

16. true

17. false

18. true

19. true

20. false

21. true

22. false

23. true

24. false

25. true

26. false

27. false

28. true

29. false

30. true

31. true

32. false

33. false

34. true

35. true

36. true

37. false

38. false

39. true

31.14 Review Questions: Multiple Choice

1. b

2. a

3. c

4. b

5. d

6. b

7. d

8. d

9. c

10. a

11. b

12. a

13. b

14. b

31.15 Review Exercises

1. Solution

Weights =

170	0
190	1
193	2
165	3
200	4

}

People

2. Solution

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

3. Solution

Names =

Toba
Issyk Kul
Baikal
Crater
Karakul

Areas =

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

Lakes

Months

0

1

2






June

July

August

Solution

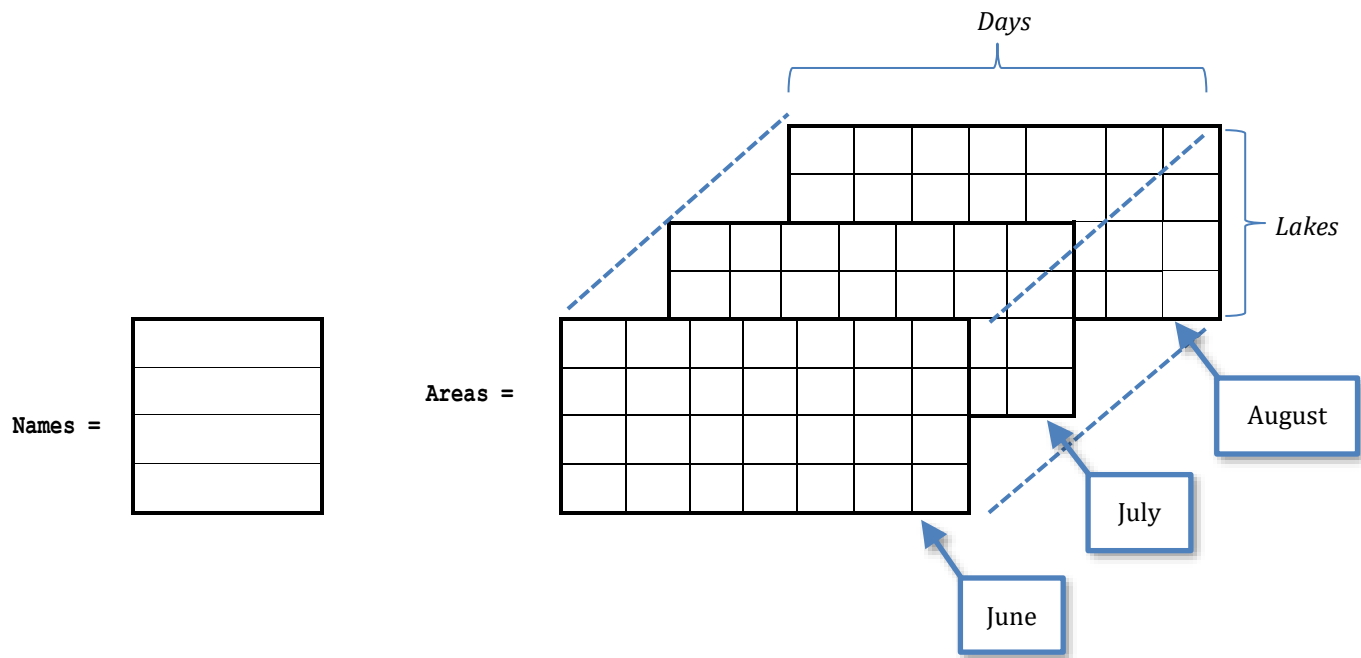
4. Solution

				<i>Dimensions</i>				
								
				<i>0</i>	<i>1</i>	<i>2</i>		
Boxes =	10	31	15	<i>0</i>				<i>Boxes</i>
	15	12	17	<i>1</i>				
	22	10	18	<i>2</i>				
	22	20	12	<i>3</i>				
	26	25	14	<i>4</i>				
	66	26	21	<i>5</i>				
	54	34	24	<i>6</i>				
	64	28	22	<i>7</i>				
	34	12	18	<i>8</i>				
	33	10	10	<i>9</i>				
								
Width		Height		Length				

5. Solution

Names =	Toba	Areas =	440	Depths =	1660	0	Lakes
	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



7. Solution

Step	Statement	\$x	\$a[0]	\$a[1]	\$a[2]
1	\$a = []	?	?	?	?
2	\$a[2] = 1	?	?	?	1
3	\$x = 0	0	?	?	1
4	\$a[\$x + \$a[2]] = 4	0	?	4	1
5	\$a[\$x] = \$a[\$x + 1] * 4	0	16	4	1

8. Solution

Step	Statement	\$x	\$a[0]	\$a[1]	\$a[2]	\$a[3]	\$a[4]
1	\$a = []	?	?	?	?	?	?
2	\$a[1] = 5	?	?	5	?	?	?
3	\$x = 0	0	?	5	?	?	?
4	\$a[\$x] = 4	0	4	5	?	?	?
5	\$a[\$a[0]] = \$a[\$x + 1] % 3	0	4	5	?	?	2
6	\$a[\$a[0] / 2] = 10	0	4	5	10	?	2
7	\$x += 2	2	4	5	10	?	2
8	\$a[\$x + 1] = \$a[\$x] + 9	2	4	5	10	19	2

9. Solution

For input value of 3

Step	Statement	\$x	\$a[0]	\$a[1]	\$a[2]	\$a[3]
1	\$a = []	?	?	?	?	?
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 = []	?	?	?	?	?
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 = []	?	?	?	?	?
2	\$a[1] = trim(fgets(STDIN))	?	?	1	?	?
3	\$x = 0	0	?	1	?	?

4	<code>\$a[\$x] = 3</code>	0	3	1	?	?
5	<code>\$a[\$a[0]] = \$a[\$x + 1] % 2</code>	0	3	1	?	3
6	<code>\$a[\$a[0] % 2] = 10</code>	0	3	10	?	3
7	<code>\$x++</code>	1	3	10	?	3
8	<code>\$a[\$x + 1] = \$a[\$x] + 9</code>	1	3	10	19	3

10. Solution

For input value of 100

Step	Statement	\$x	\$a[0]	\$a[1]	\$a[2]	\$a[3]
1	<code>\$a = []</code>	?	?	?	?	?
2	<code>\$a[1] = trim(fgets(STDIN))</code>	?	?	100	?	?
3	<code>\$x = 0</code>	0	?	100	?	?
4	<code>\$a[\$x] = 3</code>	0	3	100	?	?
5	<code>\$a[\$a[0]] = \$a[\$x + 1] % 10</code>	0	3	100	?	0
6	<code>if (\$a[3] > 5)</code>	false				
7	<code>\$a[2] = 3</code>	0	3	100	3	0

For input value of 108

Step	Statement	\$x	\$a[0]	\$a[1]	\$a[2]	\$a[3]
1	<code>\$a = []</code>	?	?	?	?	?
2	<code>\$a[1] = trim(fgets(STDIN))</code>	?	?	108	?	?
3	<code>\$x = 0</code>	0	?	108	?	?
4	<code>\$a[\$x] = 3</code>	0	3	108	?	?
5	<code>\$a[\$a[0]] = \$a[\$x + 1] % 10</code>	0	3	108	?	8
6	<code>if (\$a[3] > 5)</code>	true				
7	<code>\$a[\$a[0] % 2] = 9</code>	0	3	9	?	8
8	<code>\$x += 1</code>	1	3	9	?	8
9	<code>\$a[\$x + 1] = \$a[\$x] + 9</code>	1	3	9	18	8

For input value of 1

Step	Statement	\$x	\$a[0]	\$a[1]	\$a[2]	\$a[3]
1	<code>\$a = []</code>	?	?	?	?	?
2	<code>\$a[1] = trim(fgets(STDIN))</code>	?	?	1	?	?
3	<code>\$x = 0</code>	0	?	1	?	?
4	<code>\$a[\$x] = 3</code>	0	3	1	?	?
5	<code>\$a[\$a[0]] = \$a[\$x + 1] % 10</code>	0	3	1	?	1
6	<code>if (\$a[3] > 5)</code>	false				
7	<code>\$a[2] = 3</code>	0	3	1	3	1

11. Solution

Step	Statement	\$x	\$y	\$a[0]	\$a[1]	\$a[2]
1	\$a = []	?	?	?	?	?
2	\$x = 4	4	?	?	?	?
3	\$y = \$x - 1	4	3	?	?	?
4, 5	if (\$x > \$y) \$a[0] = 1; else \$a[0] = \$y;	4	3	1	?	?
6	\$a[1] = \$x + 3	4	3	1	7	?
7	\$y = \$y - 1	4	2	1	7	?
8	\$a[\$y] = (\$x + 5) % 2	4	2	1	7	1

12. Solution

Step	Statement	\$i	\$a[0]	\$a[1]	\$a[2]	\$a[3]	\$a[4]	\$a[5]
1	\$a = [17, 12, 45, 12, 12, 49]	?	17	12	45	12	12	49
2	\$i = 0	0	17	12	45	12	12	49
3	\$i <= 5	true						
4	if (\$a[\$i] == 12)	false						
5	\$a[\$i]++	0	18	12	45	12	12	49
6	\$i++	1	18	12	45	12	12	49
7	\$i <= 5	true						
8	if (\$a[\$i] == 12)	true						
9	\$a[\$i]--	1	18	11	45	12	12	49
10	\$i++	2	18	11	45	12	12	49
11	\$i <= 5	true						
12	if (\$a[\$i] == 12)	false						
13	\$a[\$i]++	2	18	11	46	12	12	49
14	\$i++	3	18	11	46	12	12	49
15	\$i <= 5	true						
16	if (\$a[\$i] == 12)	true						
17	\$a[\$i]--	3	18	11	46	11	12	49
18	\$i++	4	18	11	46	11	12	49
19	\$i <= 5	true						
20	if (\$a[\$i] == 12)	true						
21	\$a[\$i]--	4	18	11	46	11	11	49
22	\$i++	5	18	11	46	11	11	49
23	\$i <= 5	true						

24	if (\$a[\$i] == 12)	false						
25	\$a[\$i]++	5	18	11	46	11	11	50
26	\$i++	6	18	11	46	11	11	50
27	\$i <= 5	false						

13. Solution

Step	Statement	\$i	\$a[0]	\$a[1]	\$a[2]	\$a[3]	\$a[4]	\$a[5]
1	\$a = [10, 15, 12, 23, 22, 19]	?	10	15	12	23	22	19
2	\$i = 1	1	10	15	12	23	22	19
3	\$i <= 4	true						
4	\$a[\$i] = \$a[\$i + 1] + \$a[\$i - 1]	1	10	22	12	23	22	19
5	\$i++	2	10	22	12	23	22	19
6	\$i <= 4	true						
7	\$a[\$i] = \$a[\$i + 1] + \$a[\$i - 1]	2	10	22	45	23	22	19
8	\$i++	3	10	22	45	23	22	19
9	\$i <= 4	true						
10	\$a[\$i] = \$a[\$i + 1] + \$a[\$i - 1]	3	10	22	45	67	22	19
11	\$i++	4	10	22	45	67	22	19
12	\$i <= 4	true						
13	\$a[\$i] = \$a[\$i + 1] + \$a[\$i - 1]	4	10	22	45	67	86	19
14	\$i++	5	10	22	45	67	86	19
15	\$i <= 4	false						

14. Solution

It displays:

Navajo

Cherokee

Sioux

15. Solution

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

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

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

16. Solution

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

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

    for ($i = 0; $i <= ELEMENTS - 1; $i++) {
        $a[$i] = $a[$i] ** 2;
    }

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

17. Solution

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

    $a = [];
    for ($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], "\n";
        }
    }
?>
```

18. Solution

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

    $a = [];
    for ($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], "\n";
        }
    }
?>
```

19. Solution

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

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

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

20. Solution

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

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

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

21. Solution

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

    $a = [];
    for ($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, "\n";
?>
```

22. Solution

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

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

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

23. Solution

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

$a = [];
for ($i = 0; $i <= ELEMENTS - 1; $i++) {
    echo "Enter an integer: ";
    $a[$i] = trim(fgets(STDIN));
}

for ($i = 0; $i <= ELEMENTS - 1; $i++) {
    if ($a[$i] < 20) {
        echo $a[$i], "\n";
    }
}
?>
```

24. Solution

```
<?php
define("ELEMENTS", 60);

$a = [];
for ($i = 0; $i <= ELEMENTS - 1; $i++) {
    echo "Enter a number: ";
    $a[$i] = trim(fgets(STDIN));
}

for ($i = 0; $i <= ELEMENTS - 1; $i += 2) {
    echo $a[$i], "\n";
}
```

```
| ?>
```

25. Solution

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

    $a = [];
    for ($i = 0; $i <= ELEMENTS - 1; $i++) {
        echo "Enter a number: ";
        $a[$i] = trim(fgets(STDIN));
    }

    $total = 0;
    for ($i = 0; $i <= ELEMENTS - 1; $i += 2) {
        $total += $a[$i];
    }
    echo $total, "\n";
?>
```

26. Solution

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

    $a = [];
    for ($i = 0; $i <= ELEMENTS - 1; $i++) {
        $a[$i] = $i + 1;
    }
    ...
```

27. Solution

First approach

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

    $a = [];
    $k = 2;
    for ($i = 0; $i <= ELEMENTS - 1; $i++) {
        $a[$i] = $k;
        $k += 2;
    }
    ...
```

Second approach

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

    $a = [];
    for ($i = 0; $i <= ELEMENTS - 1; $i++) {
        $a[$i] = ($i + 1) * 2;
    }
    ...
```

28. Solution

```
| <?php
```

```

echo "Enter N: ";
$n = trim(fgets(STDIN));

$a = [];
for ($i = 1; $i <= $n; $i++) {
    $a[$i - 1] = (int)($i ** 2);
}

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

```

29. Solution

```

<?php
define("ELEMENTS", 10);

$a = [];
for ($i = 1; $i <= ELEMENTS - 1; $i++) {
    echo "Enter a number: ";
    $a[$i] = trim(fgets(STDIN));
}

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

```

30. Solution

```

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

$a = [];
for ($i = 1; $i <= ELEMENTS - 1; $i++) {
    echo "Enter a number: ";
    $a[$i] = trim(fgets(STDIN));
}

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

```

31. Solution

```

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

```



```

$a = [];
for ($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], "\n";
    }
}
?>

```

32. Solution

```

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

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

$length_limits = [0, 5, 10, 20];

for ($k = 1; $k <= 3; $k++) {
    for ($i = 0; $i <= ELEMENTS - 1; $i++) {
        if (strlen($words[$i]) >= $length_limits[$k - 1] && strlen($words[$i]) < $length_limits[$k]) {
            echo $words[$i], "\n";
        }
    }
}
?>

```

33. Solution

```

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

$a = [];
for ($i = 0; $i <= WORDS - 1; $i++) {
    echo "Enter a word: ";
    $a[$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], $j, 1) == "w") { //Alternatively use: if ($a[$i][$j] == "w")
            $count++;
        }
    }
    if ($count >= 2) {
        echo $a[$i], "\n";
    }
}

```

```
}  
?>
```

Chapter 32

32.7 Review Questions: True/False

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

32.8 Review Questions: Multiple Choice

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

32.9 Review Exercises

1. Solution

Step	Statement	\$x	\$a		
1	\$a = [[]]	?	?	?	?
			?	?	?
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 = [[]]	?	?	?	?	?
				?	?	?
2	\$i = 0	0	?	?	?	?
				?	?	?
3	\$i <= 1	true				
4	\$j = 0	0	0	?	?	?
				?	?	?

5	\$j <= 2			true		
6	\$a[\$i][\$j] = (\$i + 1) * 5 + \$j	0	0	5	?	?
				?	?	?
7	\$j++	0	1	5	?	?
				?	?	?
8	\$j <= 2	true				
9	\$a[\$i][\$j] = (\$i + 1) * 5 + \$j	0	1	5	6	?
				?	?	?
10	\$j++	0	2	5	6	?
				?	?	?
11	\$j <= 2	true				
12	\$a[\$i][\$j] = (\$i + 1) * 5 + \$j	0	2	5	6	7
				?	?	?
13	\$j++	0	3	5	6	7
				?	?	?
14	\$j <= 2	false				
15	\$i++	1	3	5	6	7
				?	?	?
16	\$i <= 1	true				
17	\$j = 0	1	0	5	6	7
				?	?	?
18	\$j <= 2	true				
19	\$a[\$i][\$j] = (\$i + 1) * 5 + \$j	1	0	5	6	7
				10	?	?
20	\$j++	1	1	5	6	7
				10	?	?
21	\$j <= 2	true				
22	\$a[\$i][\$j] = (\$i + 1) * 5 + \$j	1	1	5	6	7
				10	11	?
23	\$j++	1	2	5	6	7
				10	11	?

24	$\$j \leq 2$	true									
25	$\$a[\$i][\$j] = (\$i + 1) * 5 + \$j$	1	2	<table><tr><td>5</td><td>6</td><td>7</td></tr><tr><td>10</td><td>11</td><td>12</td></tr></table>		5	6	7	10	11	12
5	6	7									
10	11	12									
26	$\$j++$	1	3	<table><tr><td>5</td><td>6</td><td>7</td></tr><tr><td>10</td><td>11</td><td>12</td></tr></table>		5	6	7	10	11	12
5	6	7									
10	11	12									
27	$\$j \leq 2$	false									
28	$\$i++$	2	3	<table><tr><td>5</td><td>6</td><td>7</td></tr><tr><td>10</td><td>11</td><td>12</td></tr></table>		5	6	7	10	11	12
5	6	7									
10	11	12									
29	$\$i \leq 1$	false									

3. Solution

Step	Statement	\$i	\$j	\$a		
1	\$a = [[]]	?	?	?	?	?
				?	?	?
				?	?	?
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	<code>\$i++</code>	2	0	<table><tr><td>2</td><td>?</td><td>?</td></tr><tr><td>4</td><td>?</td><td>?</td></tr><tr><td>6</td><td>?</td><td>?</td></tr></table>	2	?	?	4	?	?	6	?	?
2	?	?											
4	?	?											
6	?	?											
11	<code>\$i <= 2</code>	True											
12	<code>\$a[\$i][\$j] = (\$i + 1) * 2 + \$j * 4</code>	2	0	<table><tr><td>2</td><td>?</td><td>?</td></tr><tr><td>4</td><td>?</td><td>?</td></tr><tr><td>6</td><td>?</td><td>?</td></tr></table>	2	?	?	4	?	?	6	?	?
2	?	?											
4	?	?											
6	?	?											
13	<code>\$i++</code>	3	0	<table><tr><td>2</td><td>?</td><td>?</td></tr><tr><td>4</td><td>?</td><td>?</td></tr><tr><td>6</td><td>?</td><td>?</td></tr></table>	2	?	?	4	?	?	6	?	?
2	?	?											
4	?	?											
6	?	?											
14	<code>\$i <= 2</code>	False											
15	<code>\$j++</code>	3	1	<table><tr><td>2</td><td>?</td><td>?</td></tr><tr><td>4</td><td>?</td><td>?</td></tr><tr><td>6</td><td>?</td><td>?</td></tr></table>	2	?	?	4	?	?	6	?	?
2	?	?											
4	?	?											
6	?	?											
16	<code>\$j <= 2</code>	True											
17	<code>\$i = 0</code>	0	1	<table><tr><td>2</td><td>?</td><td>?</td></tr><tr><td>4</td><td>?</td><td>?</td></tr><tr><td>6</td><td>?</td><td>?</td></tr></table>	2	?	?	4	?	?	6	?	?
2	?	?											
4	?	?											
6	?	?											
18	<code>\$i <= 2</code>	True											
19	<code>\$a[\$i][\$j] = (\$i + 1) * 2 + \$j * 4</code>	0	1	<table><tr><td>2</td><td>6</td><td>?</td></tr><tr><td>4</td><td>?</td><td>?</td></tr><tr><td>6</td><td>?</td><td>?</td></tr></table>	2	6	?	4	?	?	6	?	?
2	6	?											
4	?	?											
6	?	?											
20	<code>\$i++</code>	1	1	<table><tr><td>2</td><td>6</td><td>?</td></tr><tr><td>4</td><td>?</td><td>?</td></tr><tr><td>6</td><td>?</td><td>?</td></tr></table>	2	6	?	4	?	?	6	?	?
2	6	?											
4	?	?											
6	?	?											
21	<code>\$i <= 2</code>	True											
22	<code>\$a[\$i][\$j] = (\$i + 1) * 2 + \$j * 4</code>	1	1	<table><tr><td>2</td><td>6</td><td>?</td></tr><tr><td>4</td><td>8</td><td>?</td></tr><tr><td>6</td><td>?</td><td>?</td></tr></table>	2	6	?	4	8	?	6	?	?
2	6	?											
4	8	?											
6	?	?											
23	<code>\$i++</code>	2	1	<table><tr><td>2</td><td>6</td><td>?</td></tr><tr><td>4</td><td>8</td><td>?</td></tr><tr><td>6</td><td>?</td><td>?</td></tr></table>	2	6	?	4	8	?	6	?	?
2	6	?											
4	8	?											
6	?	?											
24	<code>\$i <= 2</code>	True											
25	<code>\$a[\$i][\$j] = (\$i + 1) * 2 + \$j * 4</code>	2	1	<table><tr><td>2</td><td>6</td><td>?</td></tr><tr><td>4</td><td>8</td><td>?</td></tr><tr><td>6</td><td>10</td><td>?</td></tr></table>	2	6	?	4	8	?	6	10	?
2	6	?											
4	8	?											
6	10	?											

26	<code>\$i++</code>	3	1	<table><tr><td>2</td><td>6</td><td>?</td></tr><tr><td>4</td><td>8</td><td>?</td></tr><tr><td>6</td><td>10</td><td>?</td></tr></table>	2	6	?	4	8	?	6	10	?
2	6	?											
4	8	?											
6	10	?											
27	<code>\$i <= 2</code>	False											
28	<code>\$j++</code>	3	2	<table><tr><td>2</td><td>6</td><td>?</td></tr><tr><td>4</td><td>8</td><td>?</td></tr><tr><td>6</td><td>10</td><td>?</td></tr></table>	2	6	?	4	8	?	6	10	?
2	6	?											
4	8	?											
6	10	?											
29	<code>\$j <= 2</code>	True											
30	<code>\$i = 0</code>	0	2	<table><tr><td>2</td><td>6</td><td>?</td></tr><tr><td>4</td><td>8</td><td>?</td></tr><tr><td>6</td><td>10</td><td>?</td></tr></table>	2	6	?	4	8	?	6	10	?
2	6	?											
4	8	?											
6	10	?											
31	<code>\$i <= 2</code>	True											
32	<code>\$a[\$i][\$j] = (\$i + 1) * 2 + \$j * 4</code>	0	2	<table><tr><td>2</td><td>6</td><td>10</td></tr><tr><td>4</td><td>8</td><td>?</td></tr><tr><td>6</td><td>10</td><td>?</td></tr></table>	2	6	10	4	8	?	6	10	?
2	6	10											
4	8	?											
6	10	?											
33	<code>\$i++</code>	1	2	<table><tr><td>2</td><td>6</td><td>10</td></tr><tr><td>4</td><td>8</td><td>?</td></tr><tr><td>6</td><td>10</td><td>?</td></tr></table>	2	6	10	4	8	?	6	10	?
2	6	10											
4	8	?											
6	10	?											
34	<code>\$i <= 2</code>	True											
35	<code>\$a[\$i][\$j] = (\$i + 1) * 2 + \$j * 4</code>	1	2	<table><tr><td>2</td><td>6</td><td>10</td></tr><tr><td>4</td><td>8</td><td>12</td></tr><tr><td>6</td><td>10</td><td>?</td></tr></table>	2	6	10	4	8	12	6	10	?
2	6	10											
4	8	12											
6	10	?											
36	<code>\$i++</code>	2	2	<table><tr><td>2</td><td>6</td><td>10</td></tr><tr><td>4</td><td>8</td><td>12</td></tr><tr><td>6</td><td>10</td><td>?</td></tr></table>	2	6	10	4	8	12	6	10	?
2	6	10											
4	8	12											
6	10	?											
37	<code>\$i <= 2</code>	True											
38	<code>\$a[\$i][\$j] = (\$i + 1) * 2 + \$j * 4</code>	2	2	<table><tr><td>2</td><td>6</td><td>10</td></tr><tr><td>4</td><td>8</td><td>12</td></tr><tr><td>6</td><td>10</td><td>14</td></tr></table>	2	6	10	4	8	12	6	10	14
2	6	10											
4	8	12											
6	10	14											
39	<code>\$i++</code>	3	2	<table><tr><td>2</td><td>6</td><td>10</td></tr><tr><td>4</td><td>8</td><td>12</td></tr><tr><td>6</td><td>10</td><td>14</td></tr></table>	2	6	10	4	8	12	6	10	14
2	6	10											
4	8	12											
6	10	14											
40	<code>\$i <= 2</code>	False											
41	<code>\$j++</code>	3	3	<table><tr><td>2</td><td>6</td><td>10</td></tr><tr><td>4</td><td>8</td><td>12</td></tr><tr><td>6</td><td>10</td><td>14</td></tr></table>	2	6	10	4	8	12	6	10	14
2	6	10											
4	8	12											
6	10	14											

42 $\$j \leq 2$

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);

$a = [[]];
for ($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, "\n";
        }
    }
}
?>
```

10. Solution

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

$a = [[]];
for ($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], "\n";
    }
}
?>
```

11. Solution

```
<?php
define("ROWS", 12);
```

```

define("COLUMNS", 8);

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

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

```

12. Solution

```

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

$a = [];
for ($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, "\n";
?>

```

13. Solution

```

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

$a = [];
for ($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);

$a = [[]];
for ($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 = [[]];

```

```

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] == (int)$a[$i][$j]) {
            echo $i, ", ", $j, "\n";
        }
    }
}
?>

```

16. Solution

```

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

$a = [[]];
for ($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, "\n";
?>

```

17. Solution

```

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

$a = [[]];
for ($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);

$a = [[]];
for ($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], "\n";
        }
    }
}
?>

```

19. Solution

First approach

```

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

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

$length_limits = [5, 10, 20];

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

```

```
?>
```

Second approach

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

$a = [];
for ($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 * 2 ** $k) {
                echo $a[$i][$j], "\n";
            }
        }
    }
}
?>
```

Chapter 33

33.7 Review Questions: True/False

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

33.8 Review Questions: Multiple Choice

- | | |
|------|------|
| 1. a | 4. b |
| 2. b | 5. a |
| 3. c | 6. a |

33.9 Review Exercises**1. Solution**

```

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

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

$average = [];
for ($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\n";
    }
    elseif ($average[$i] < 70) {
        echo "D\n";
    }
    elseif ($average[$i] < 80) {
        echo "C\n";
    }
}

```



```

    elseif ($average[$i] < 90) {
        echo "B\n";
    }
    else {
        echo "A\n";
    }
}
?>

```

2. Solution

```

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

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

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

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

```

3. Solution

```

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

```

```

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

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

```

4. Solution

```

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

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

```

5. Solution

```

<?php
define("PLAYERS", 24);

```

```

define("MATCHES", 10);

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

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

```

6. Solution

```

<?php
define("STUDENTS", 12);
define("LESSONS", 6);

$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));
    }
}

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

```

```

for ($j = 0; $j <= LESSONS - 1; $j++) {
    $total = 0;
    for ($i = 0; $i <= STUDENTS - 1; $i++) {
        $total += $grades[$i][$j];
    }
    echo $total / STUDENTS, "\n";
}

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
define("ARTISTS", 15);
define("JUDGES", 5);

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

$artist_names = [];
$song_titles = [];
$score = [[]];
for ($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];
        echo " gotten from judge ", $judge_names[$j], ": ";
        $score[$i][$j] = trim(fgets(STDIN));
    }
}

for ($i = 0; $i <= ARTISTS - 1; $i++) {
    $total = 0;
    for ($j = 0; $j <= JUDGES - 1; $j++) {
        $total += $score[$i][$j];
    }
}

```

```

    }
    echo $artist_names[$i], ", ", $song_titles[$i], ": ", $total, "\n";
}

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

```

8. Solution

```

<?php
define("PEOPLE", 30);
define("MONTHS", 12);

$weights = [[]];
$heights = [[]];
for ($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, "\n";
    echo $average_weight * 702 / $average_height ** 2, "\n";
}

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

```

9. Solution

```

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

```

```

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

$total = 0;
for ($i = 0; $i <= CONSUMERS - 1; $i++) {
    $consumed = $meter_reading[$i][1] - $meter_reading[$i][0];
    echo $consumed, "\n";
    $payment = $consumed * 0.07;
    $payment += VAT * $payment;
    echo $payment, "\n";

    $total += $consumed;
}

echo $total, ", ", ($total * 0.07 + $total * 0.07 * VAT), "\n";
?>

```

10. Solution

```

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

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

$currency = ["British Pounds Sterling", "Euros", "Canadian Dollars", "Australian Dollars"];

$rate = [[1.320, 1.321, 1.332, 1.331, 1.341],
         [1.143, 1.156, 1.138, 1.122, 1.129],
         [0.757, 0.764, 0.760, 0.750, 0.749],
         [0.720, 0.725, 0.729, 0.736, 0.739]
        ];

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

```

11. Solution

```

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

```

```

$days = ["Monday", "Tuesday", "Wednesday", "Thursday", "Friday"];

$pay_rate = trim(fgets(STDIN));

$names = [];
$hours_worked_per_day = [[]];
for ($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));
    }
}

$hours_worked_per_week = [];
for ($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], "\n";
    }
}

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, "\n";
}

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!\n";
            }
        }
    }
}

for ($j = 0; $j <= DAYS - 1; $j++) {
    $total = 0;
    for ($i = 0; $i <= EMPLOYEES - 1; $i++) {
        if ($hours_worked_per_day[$i][$j] <= 8) {
            $gross_pay = $pay_rate * $hours_worked_per_day[$i][$j];
        }
        else {
            $gross_pay = $pay_rate * 8 + 1.5 * $pay_rate * ($hours_worked_per_day[$i][$j] - 8);
        }
    }
}

```

```

    }
    $total += $gross_pay;
}
echo $days[$j], " ", $total, "\n";
}
?>

```

12. Solution

First approach

```

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

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

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

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

```

Second approach

```

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

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

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

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

```


13. Solution

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

$a = [16, 12, 3, 5, 6, 9, 18, 19, 20];

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

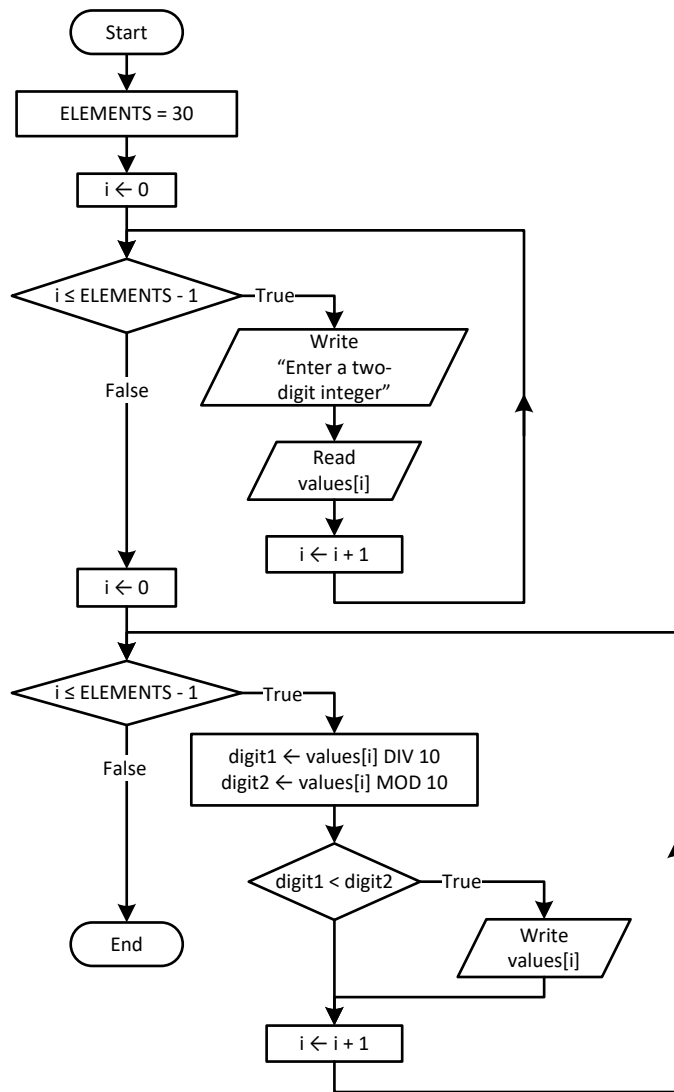
for ($i = 0; $i <= ROWS - 1; $i++) {
    for ($j = 0; $j <= COLUMNS - 1; $j++) {
        echo $b[$i][$j], "\t";
    }
    echo "\n";
}
?>
```


34.7 Review Questions: True/False**Chapter 34**

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

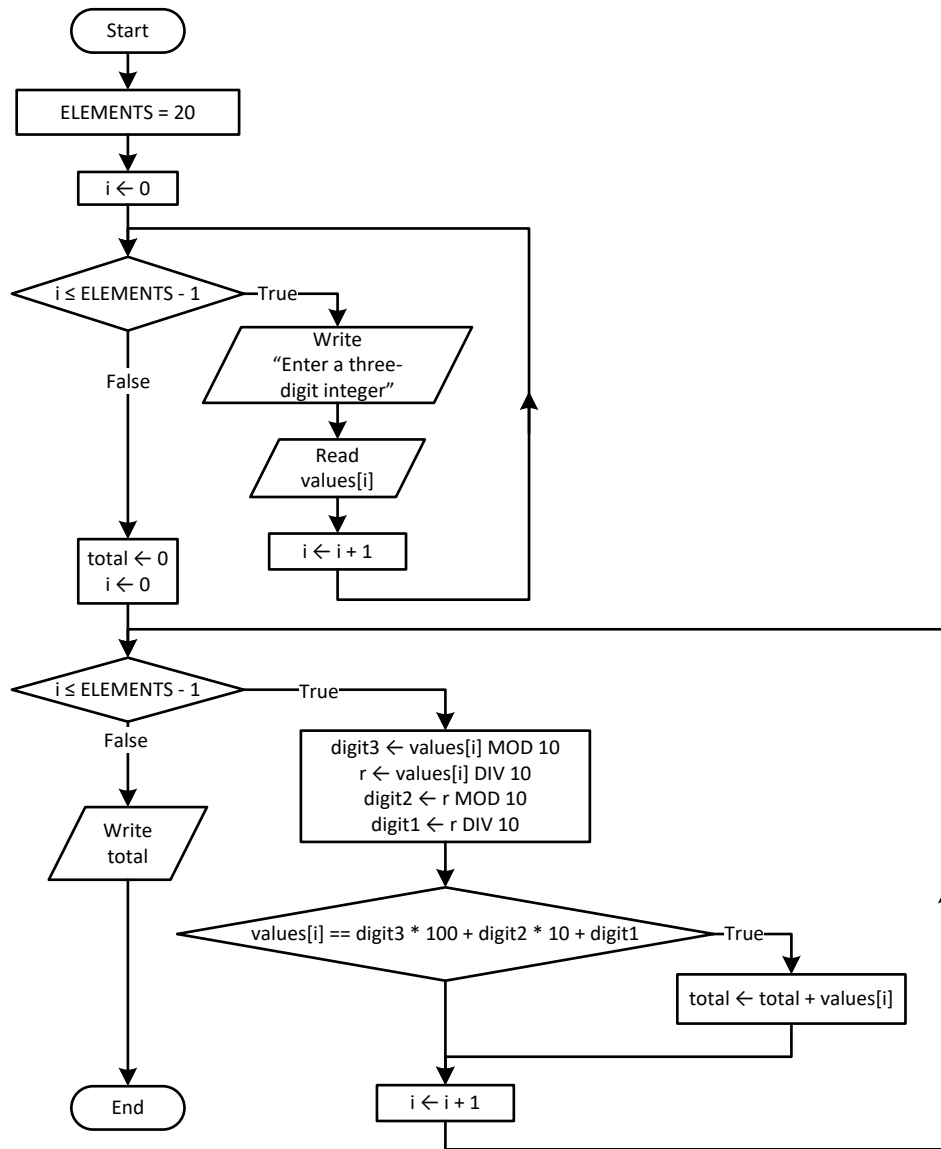
34.8 Review Exercises

1. Solution

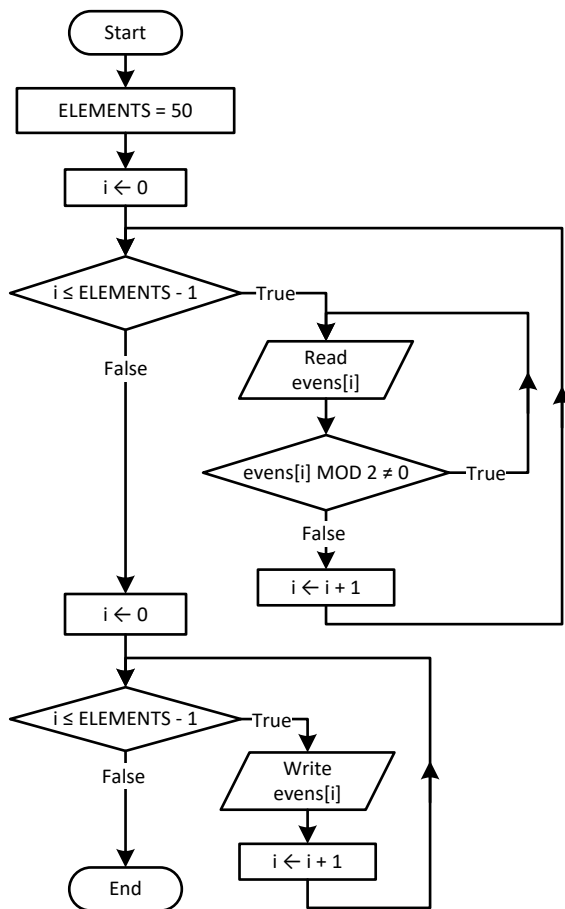


 Please note that since flowcharts are a loose method to represent an algorithm, it is not necessary to initialize an array within a flowchart; that is, there is no need to represent the statement `values = []`.

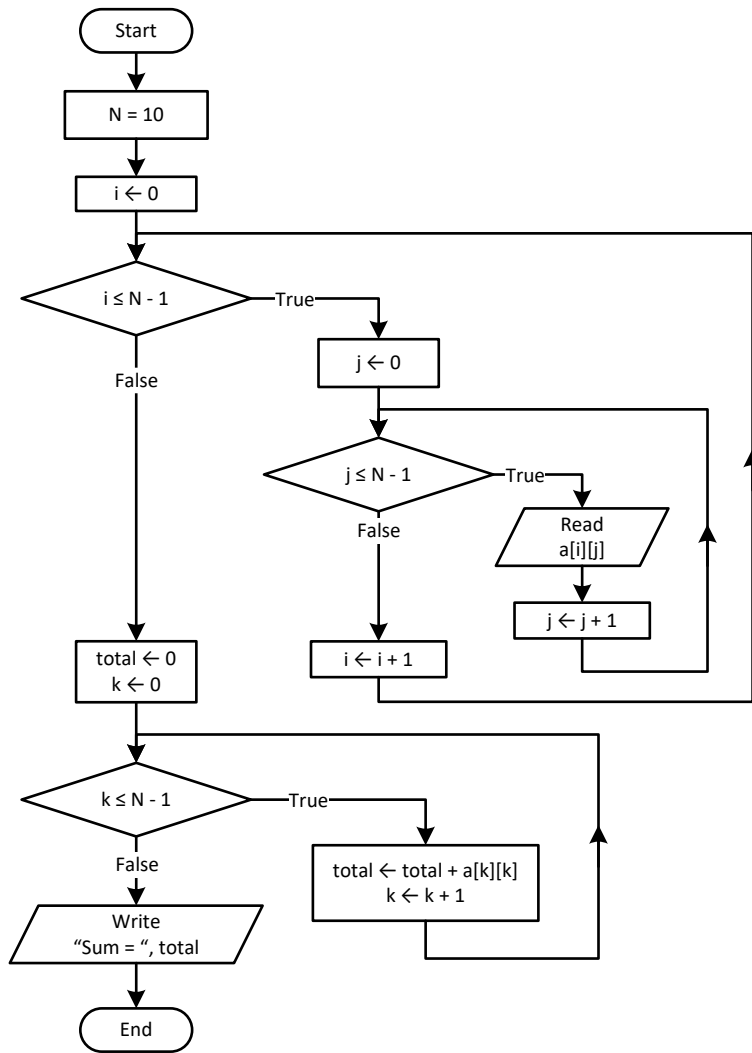
2. Solution



3. Solution



4. Solution



5. Solution

```

for ($i = 0; $i <= CITIES - 1; $i++){
    do {
        $b[$i] = trim(fgets(STDIN));
    } while ($b[$i] >= 0);
}

```

6. Solution

```

<?php
$pos = [];
$neg = [];

$i = 1;
$m = 0;
$n = 0;
do {

```

```

    $b = trim(fgets(STDIN));
    if ($b < 0) {
        $pos[$m] = $b;
        $m++;
    }
    else {
        $neg[$n] = $b;
        $n++;
    }
    $i++;
} while ($i < 90);
echo "The End\n";
?>

```

7. Solution

```

$max_i = 0;
$max_j = 0;
for ($i = 0; $i <= CITIES - 1; $i++) {
    for ($j = 0; $j <= CITIZENS - 1; $j++) {
        if ($a[$i][$j] > $a[$max_i][$max_j]) {
            $max_i = $i;
            $max_j = $j;
        }
    }
}

echo $a[$max_i][$max_j], "\n";

```

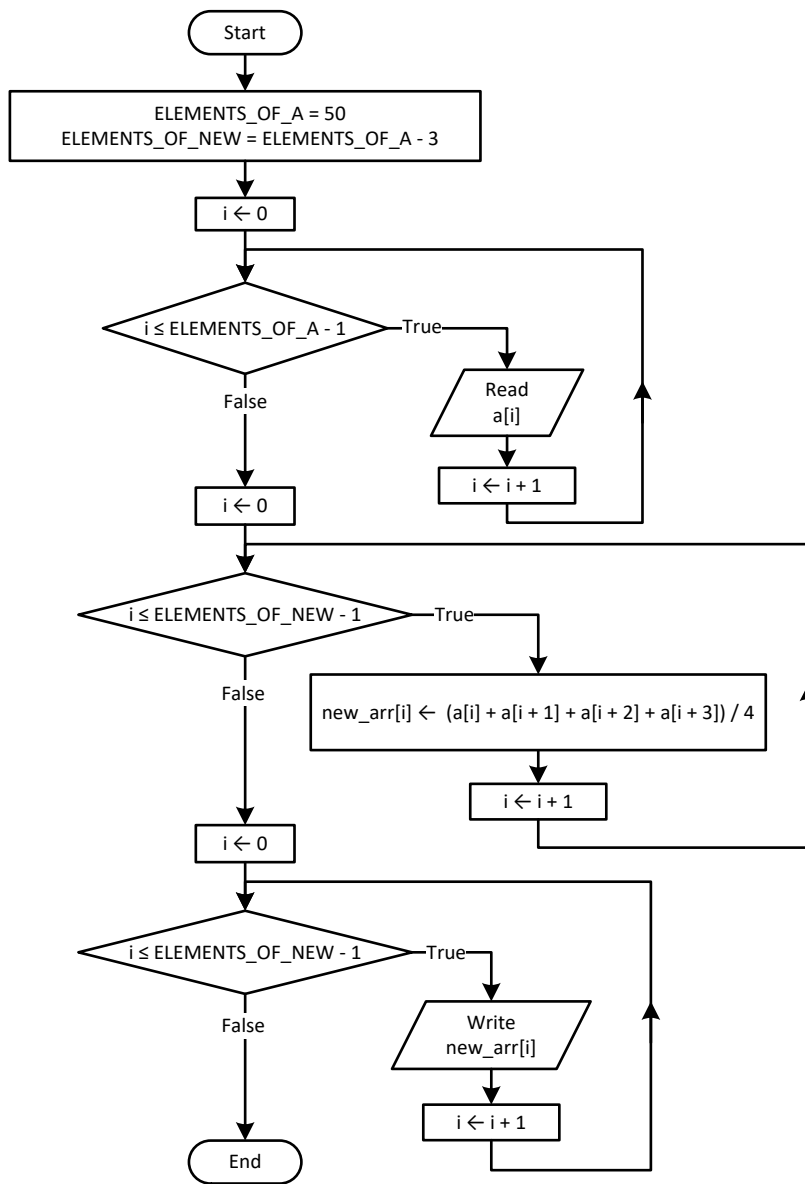
8. Solution

```

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\n";
            $a[$i][$j] = trim(fgets(STDIN));
        }
    }
}

```

9. Solution



```

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

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

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

for ($i = 0; $i <= ELEMENTS_OF_NEW - 1; $i++) {

```



```
        echo $new_arr[$i], "\t\n";
    }
?>
```

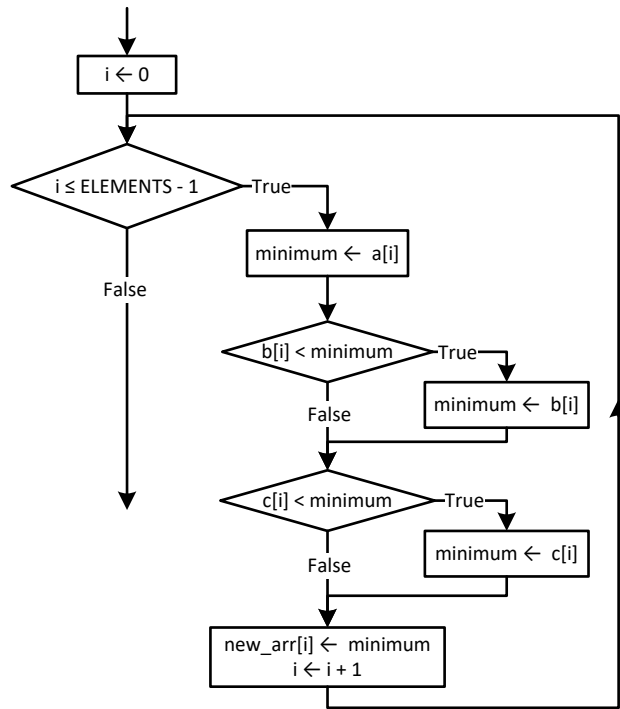
10. Solution

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

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

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

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



11. Solution

```

<?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);

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

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

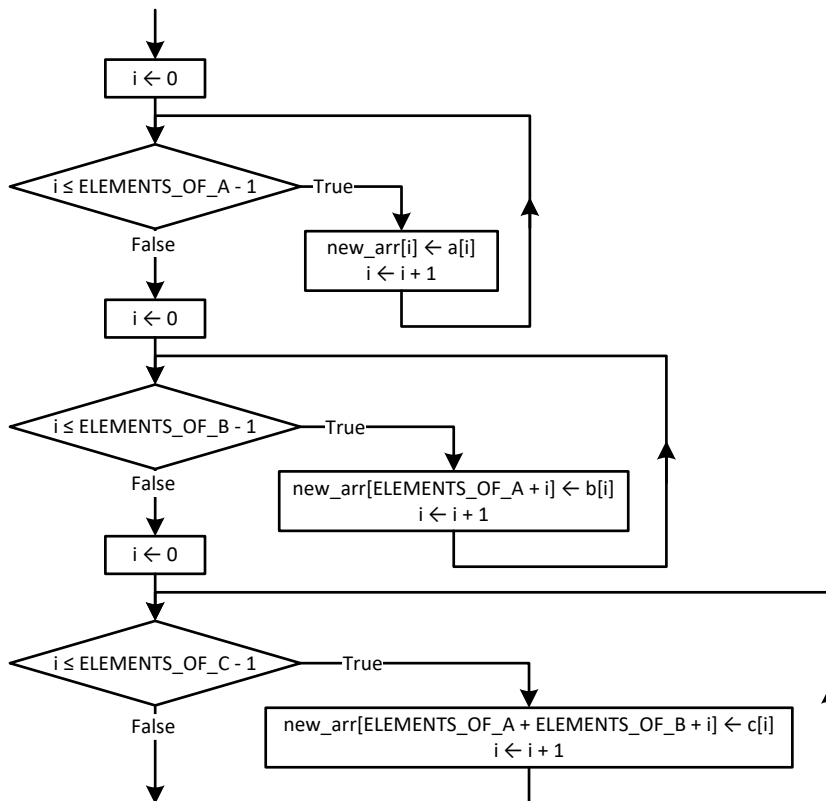
```

```

    $new_arr[ELEMENTS_OF_B + ELEMENTS_OF_C + $i] = $a[$i];
}

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

```



12. 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);

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

$b = [];
for ($i = 0; $i <= ROWS - 1; $i++) {

```

```

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

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

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

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

```

13. Solution

```

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

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

$reals = [];
$integers = [];
$reals_index = 0;
$integers_index = 0;
for ($i = 0; $i <= ELEMENTS - 1; $i++) {

```

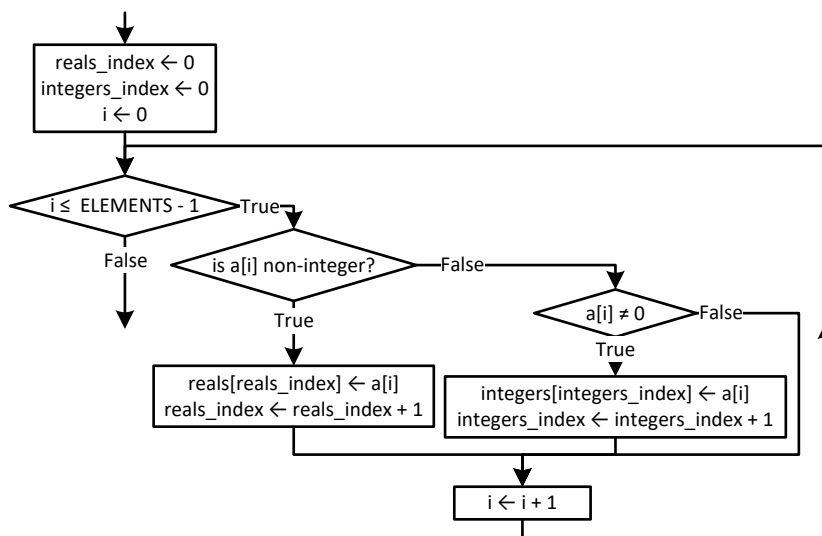
```

    if ($a[$i] != (int)$a[$i]) {
        $reals[$reals_index] = $a[$i];
        $reals_index++;
    }
    elseif ($a[$i] != 0) {
        $integers[$integers_index] = (int)$a[$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";
}
?>

```



14. Solution

```

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

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

$b = [];
$k = 0;
for ($i = 0; $i <= ELEMENTS - 1; $i++) {
    $digit3 = $a[$i] % 10;
    $r = (int)($a[$i] / 10);
}

```

```

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

    if ($digit1 < $digit2 && $digit2 < $digit3) {
        $b[$k] = $a[$i];
        $k++;
    }
}

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

```

15. Solution

```

<?php
define("PRODUCTS", 10);
define("CITIZENS", 200);

$prod_names = [];
$answers = [[]];
for ($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") > 0) {
            echo "Error! \n";
            $answers[$i][$j] = trim(fgets(STDIN));
        }
    }
}

$count_A = [];
for ($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], "\n";
}

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, "\n";

```

```

}

$maximum = $count_A[0];
for ($i = 1; $i <= PRODUCTS - 1; $i++) {
    if ($count_A[$i] > $maximum) {
        $maximum = $count_A[$i];
    }
}
for ($i = 0; $i <= PRODUCTS - 1; $i++) {
    if ($count_A[$i] == $maximum) {
        echo $prod_names[$i], "\n";
    }
}
}
?>

```

16. Solution

```

<?php
define("US_CITIES", 20);
define("CANADIAN_CITIES", 20);

$sus_names = [];
for ($i = 0; $i <= US_CITIES - 1; $i++) {
    echo "Enter name for US city No ", ($i + 1), ": \n";
    $sus_names[$i] = trim(fgets(STDIN));
}

$canadian_names = [];
for ($j = 0; $j <= CANADIAN_CITIES - 1; $j++) {
    echo "Enter name for Canadian city No ", ($j + 1), ": \n";
    $canadian_names[$j] = trim(fgets(STDIN));
}

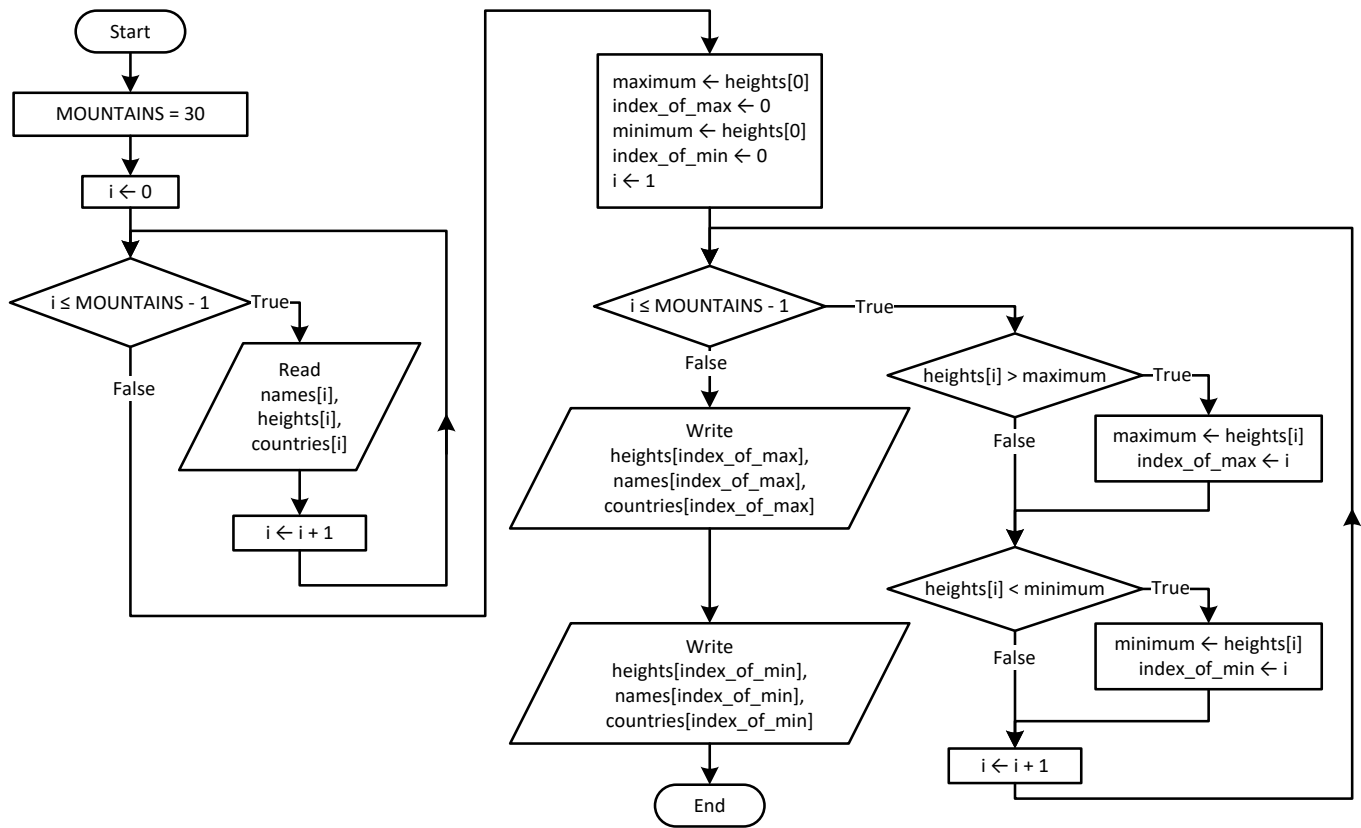
$distances = [[]];
for ($i = 0; $i <= US_CITIES - 1; $i++) {
    for ($j = 0; $j <= CANADIAN_CITIES - 1; $j++) {
        echo "Enter distance between ", $sus_names[$i], " and ", $canadian_names[$j], ": \n";
        $distances[$i][$j] = trim(fgets(STDIN));
    }
}

for ($i = 0; $i <= US_CITIES - 1; $i++) {
    $minimum = $distances[$i][0];
    $min_j = 0;
    for ($j = 1; $j <= CANADIAN_CITIES - 1; $j++) {
        if ($distances[$i][$j] < $minimum) {
            $minimum = $distances[$i][$j];
            $min_j = $j;
        }
    }
    echo "Closest Canadian city to ", $sus_names[$i], " is ", $canadian_names[$min_j], "\n";
}

```

?>

17. Solution



```

<?php
define("MOUNTAINS", 30);

$names = [];
$heights = [];
$countries = [];
for ($i = 0; $i <= MOUNTAINS - 1; $i++) {
    $names[$i] = trim(fgets(STDIN));
    $heights[$i] = trim(fgets(STDIN));
    $countries[$i] = trim(fgets(STDIN));
}

$maximum = $heights[0];
$index_of_max = 0;
$minimum = $heights[0];
$index_of_min = 0;
for ($i = 1; $i <= MOUNTAINS - 1; $i++) {
    if ($heights[$i] > $maximum) {
        $maximum = $heights[$i];
        $index_of_max = $i;
    }
    if ($heights[$i] < $minimum) {
        $minimum = $heights[$i];
    }
}
  
```



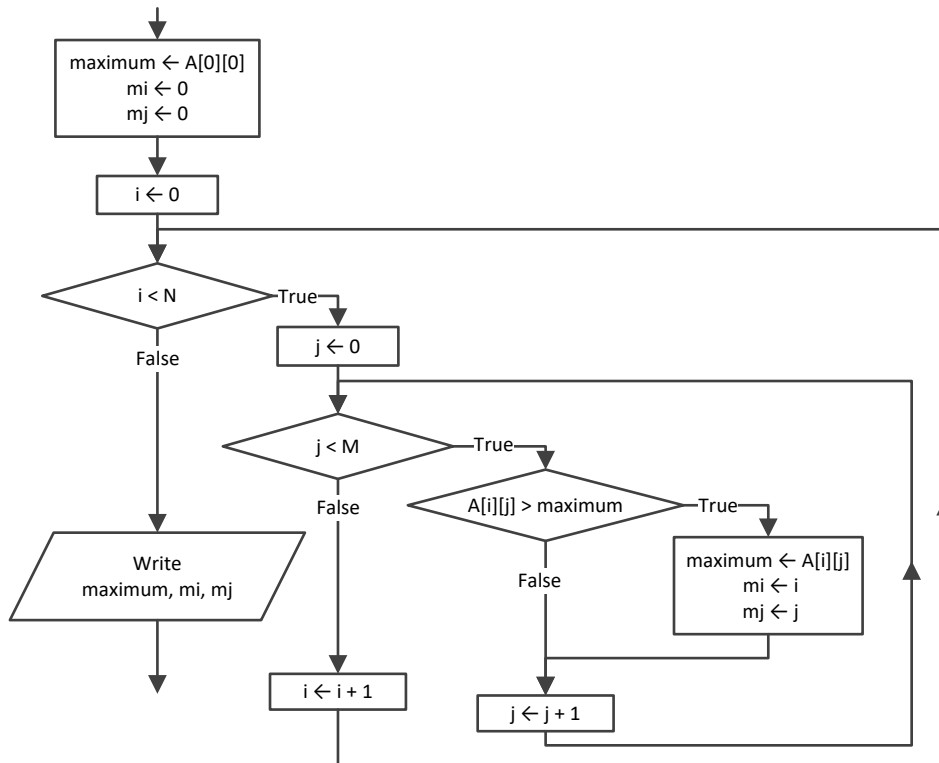
```

    $index_of_min = $i;
  }
}

echo $heights[$index_of_max], " ", $names[$index_of_max], " ", $countries[$index_of_max], "\n";
echo $heights[$index_of_min], " ", $names[$index_of_min], " ", $countries[$index_of_min], "\n";
?>

```

18. Solution



19. Solution

```

<?php
define("TEAMS", 26);
define("GAMES", 15);

$names = [];
$results = [[]];
for ($i = 0; $i <= TEAMS - 1; $i++) {
    $names[$i] = trim(fgets(STDIN));
    for ($j = 0; $j <= GAMES - 1; $j++) {
        $results[$i][$j] = trim(fgets(STDIN));
    }
}

$points = [];
for ($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;
        }
    }
}

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

echo $names[$m_i], "\n";
?>

```

20. Solution

```

<?php
define("OBJECTS", 10);
define("FALLS", 20);

$heights = [[]];
$times = [[]];
for ($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));
    }
}

$g = [[]];
for ($i = 0; $i <= OBJECTS - 1; $i++) {
    for ($j = 0; $j <= FALLS - 1; $j++) {
        $g[$i][$j] = 2 * $heights[$i][$j] / $times[$i][$j] ** 2;
    }
}

$minimum = [];
$maximum = [];
for ($i = 0; $i <= OBJECTS - 1; $i++) {
    $minimum[$i] = $g[$i][0];
    $maximum[$i] = $g[$i][0];
    for ($j = 1; $j <= FALLS - 1; $j++) {
        if ($g[$i][$j] < $minimum[$i]) {
            $minimum[$i] = $g[$i][$j];
        }
    }
}

```

```

    }
    if ($g[$i][$j] > $maximum[$i]) {
        $maximum[$i] = $g[$i][$j];
    }
}
}

for ($i = 0; $i <= OBJECTS - 1; $i++) {
    echo $minimum[$i], ", ", $maximum[$i], "\n";
}

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

echo $mini, ", ", $maxi, "\n";
?>

```

21. Solution

```

<?php
define("STATIONS", 10);
define("DAYS", 365);

$names = [];
$co2 = [[]];
for ($i = 0; $i <= STATIONS - 1; $i++) {
    $names[$i] = trim(fgets(STDIN));
    for ($j = 0; $j <= DAYS - 1; $j++) {
        $co2[$i][$j] = trim(fgets(STDIN));
    }
}

$average = [];
for ($i = 0; $i <= STATIONS - 1; $i++) {
    $average[$i] = 0;
    for ($j = 0; $j <= DAYS - 1; $j++) {
        $average[$i] += $co2[$i][$j];
    }
    $average[$i] /= DAYS;
}

$minimum = $average[0];
$m_i = 0;
for ($i = 1; $i <= STATIONS - 1; $i++) {
    if ($average[$i] < $minimum) {

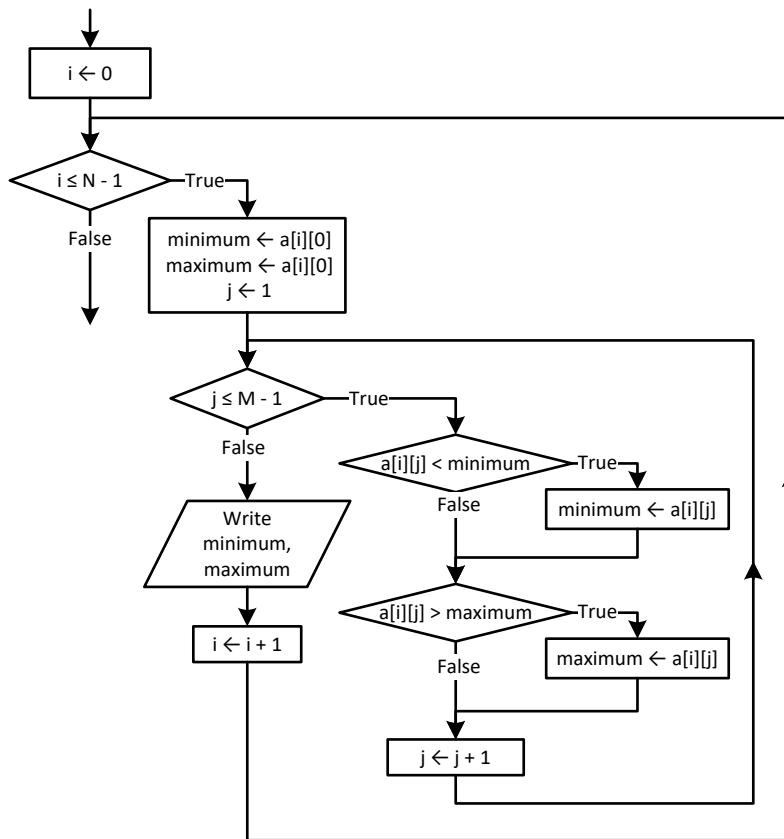
```

```

    $minimum = $average[$i];
    $m_i = $i;
  }
}
echo $names[$m_i], "\n";
?>

```

22. Solution



23. Solution

First approach

```

<?php
define("ROWS", 20);
define("COLUMNS", 30);

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

$minimum = [];
$maximum = [];
for ($j = 0; $j <= COLUMNS - 1; $j++) {

```

```

$minimum[$j] = $b[0][$j];
$maximum[$j] = $b[0][$j];
for ($i = 1; $i <= ROWS - 1; $i++) {
    if ($b[$i][$j] < $minimum[$j]) {
        $minimum[$j] = $b[$i][$j];
    }
    if ($b[$i][$j] > $maximum[$j]) {
        $maximum[$j] = $b[$i][$j];
    }
}
}

for ($j = 0; $j <= COLUMNS - 1; $j++) {
    echo $minimum[$j], " ", $maximum[$j], "\n";
}
?>

```

Second approach

```

<?php
define("ROWS", 20);
define("COLUMNS", 30);

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

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

```

24. Solution

```

<?php
define("TEAMS", 20);
define("GAMES", 10);

$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 for team ", $names[$i], " for game No ", ($j + 1), ": \n";
    $results[$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));
    }
}
}

$points = [];
for ($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) break;
}

echo "Gold: ", $names[0], "\n";
echo "Silver: ", $names[1], "\n";
echo "Bronze: ", $names[2], "\n";
?>

```

25. Solution

```

<?php
define("PEOPLE", 50);

```

```

$names = [];
$heights = [];
for ($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";
}
?>

```

26. Solution

```

<?php
define("ARTISTS", 12);
define("JUDGES", 10);

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

```

```

$total = [];
for ($i = 0; $i <= ARTISTS - 1; $i++) {
    $total[$i] = 0;
    for ($j = 1; $j <= JUDGES - 1; $j++) {
        $total[$i] += $score[$i][$j];
    }
}

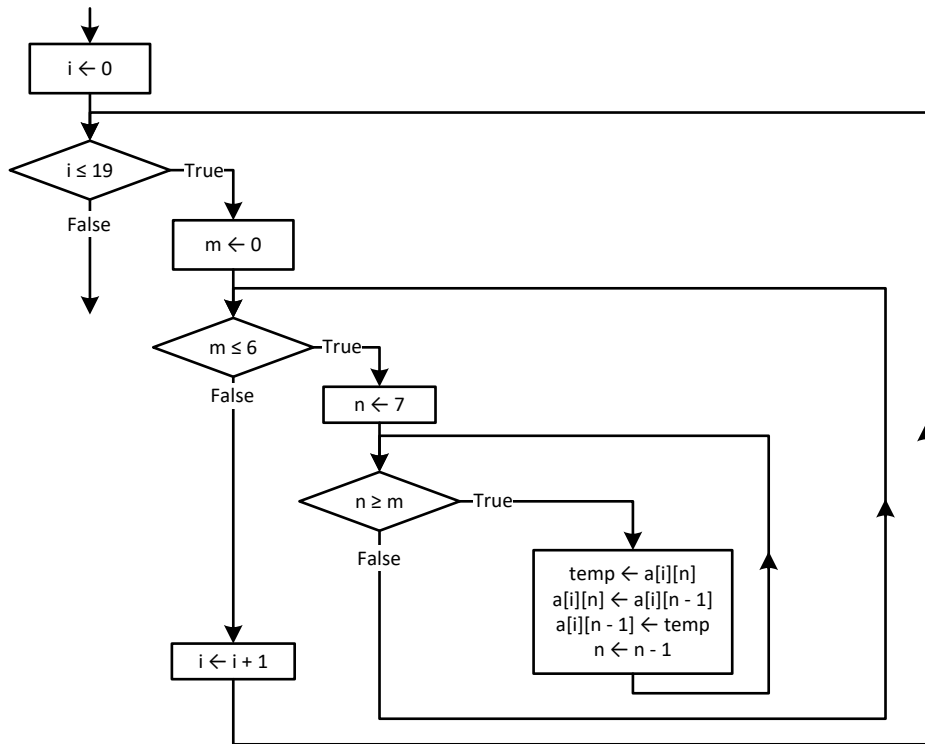
for ($i = 0; $i <= ARTISTS - 1; $i++) {
    $minimum = $score[$i][0];
    $maximum = $score[$i][0];
    for ($j = 1; $j <= JUDGES - 1; $j++) {
        if ($score[$i][$j] < $minimum) {
            $minimum = $score[$i][$j];
        }
        if ($score[$i][$j] > $maximum) {
            $maximum = $score[$i][$j];
        }
    }
    $total[$i] = $total[$i] - $minimum - $maximum;
    echo $total[$i], "\n";
}

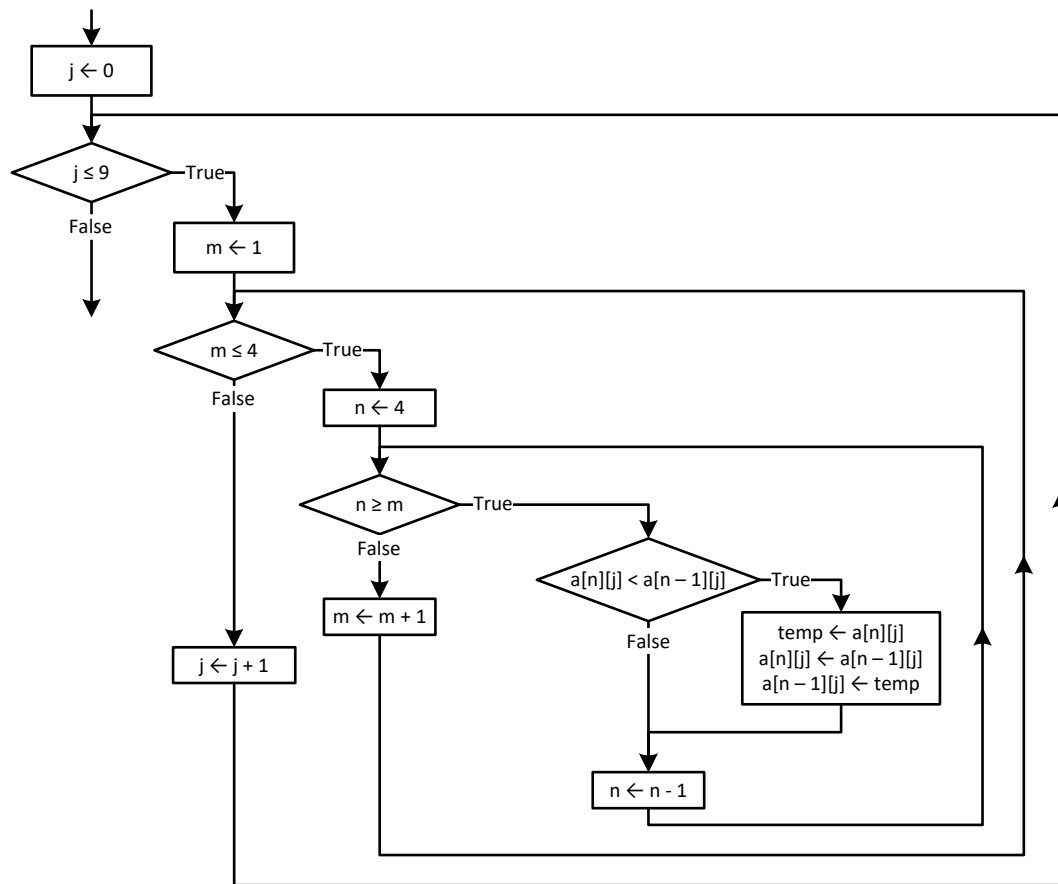
for ($m = 1; $m <= ARTISTS - 1; $m++) {
    for ($n = ARTISTS - 1; $n >= $m; $n--) {
        if ($total[$n] > $total[$n - 1]) {
            $temp = $total[$n];
            $total[$n] = $total[$n - 1];
            $total[$n - 1] = $temp;

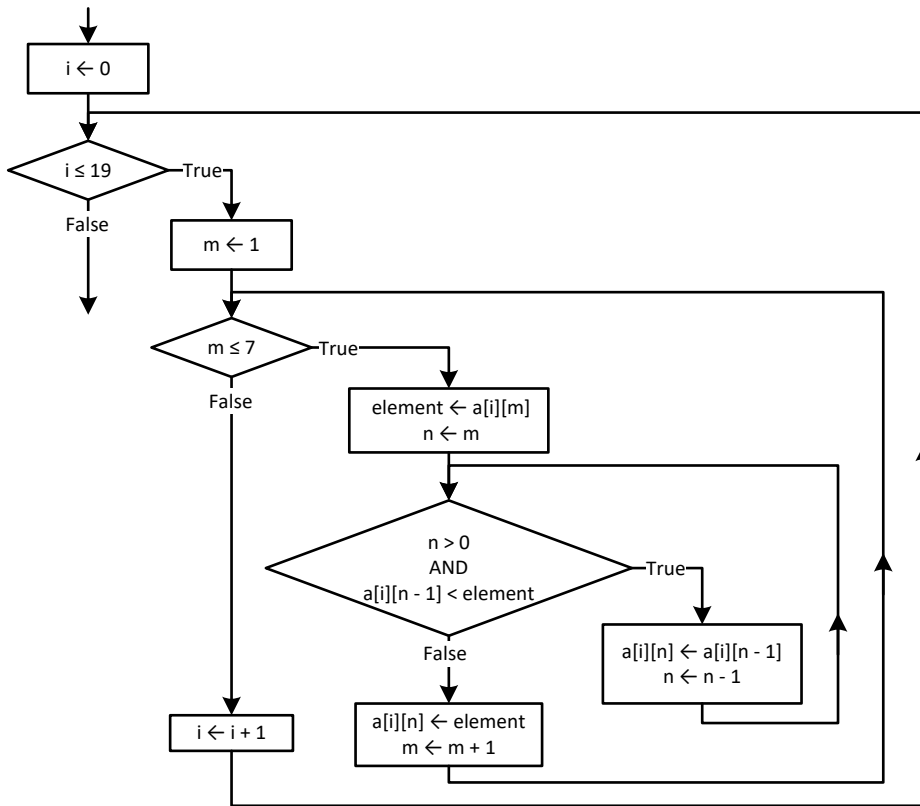
            $temp = $artist_names[$n];
            $artist_names[$n] = $artist_names[$n - 1];
            $artist_names[$n - 1] = $temp;
        }
        elseif ($total[$n] == $total[$n - 1]) {
            if ($artist_names[$n] < $artist_names[$n - 1]) {
                $temp = $artist_names[$n];
                $artist_names[$n] = $artist_names[$n - 1];
                $artist_names[$n - 1] = $temp;
            }
        }
    }
}

for ($i = 0; $i <= ARTISTS - 1; $i++) {
    echo $artist_names[$i], ", ", $total[$i], "\n";
}
?>

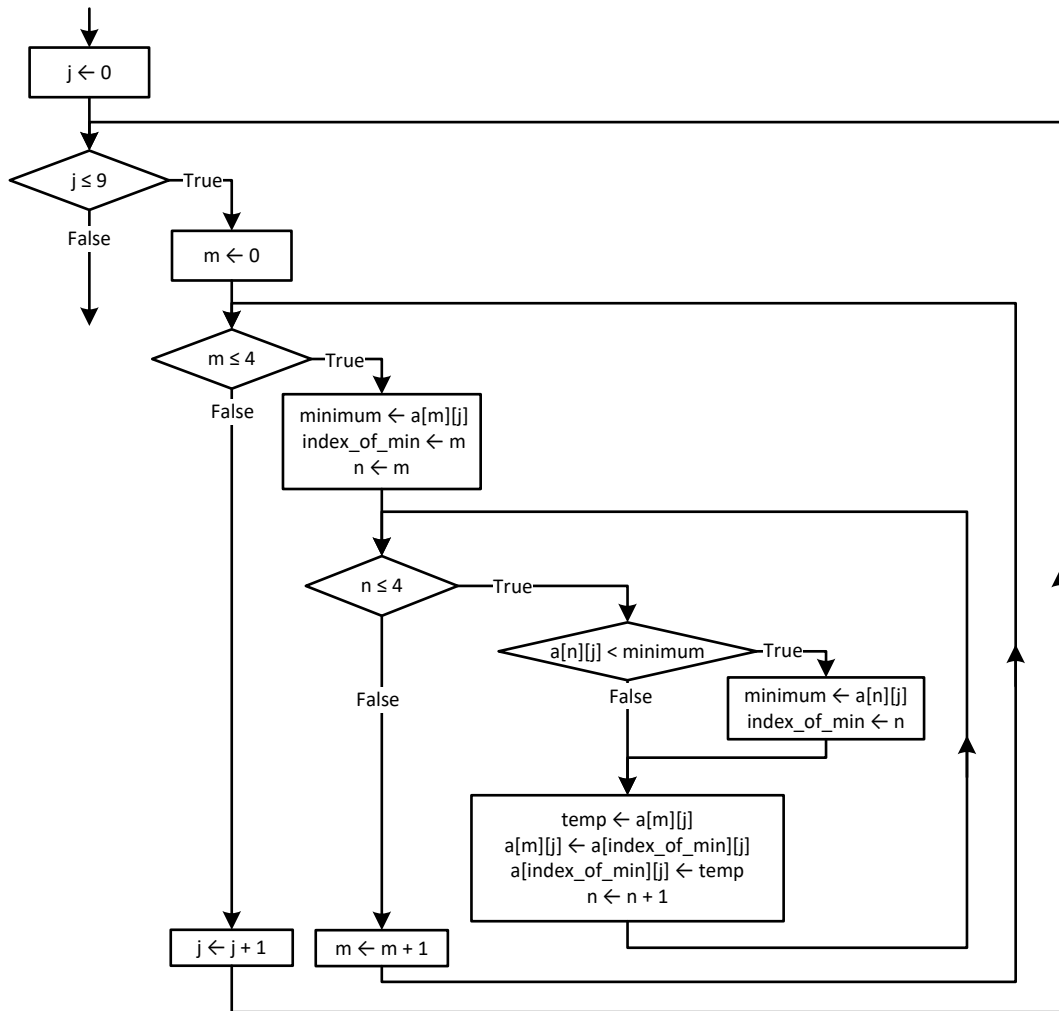
```


27. Solution

28. Solution

29. Solution

30. Solution



31. Solution

```

<?php
define("PEOPLE", 10);
define("PUZZLES", 8);

$names = [];
$times = [[]];
for ($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++) {
        $minimum = $times[$i][$m];
    }
}
  
```

```

    $index_of_min = $m;
    for ($n = $m; $n <= PUZZLES - 1; $n++) {
        if ($times[$i][$n] < $minimum) {
            $minimum = $times[$i][$n];
            $index_of_min = $n;
        }
    }
    $temp = $times[$i][$m];
    $times[$i][$m] = $times[$i][$index_of_min];
    $times[$i][$index_of_min] = $temp;
}
}

for ($i = 0; $i <= PEOPLE - 1; $i++) {
    echo $names[$i], "\n";
    for ($j = 0; $j <= 2; $j++) {
        echo $times[$i][$j], "\n";
    }
}

$average = [];
for ($i = 0; $i <= PEOPLE - 1; $i++) {
    $average[$i] = 0;
    for ($j = 0; $j <= PUZZLES - 1; $j++) {
        $average[$i] += $times[$i][$j];
    }
    $average[$i] /= PUZZLES;
}

for ($m = 0; $m <= PEOPLE - 1; $m++) {
    $minimum = $average[$m];
    $index_of_min = $m;
    for ($n = $m; $n <= PEOPLE - 1; $n++) {
        if ($average[$n] < $minimum) {
            $minimum = $average[$n];
            $index_of_min = $n;
        }
    }
    $temp = $average[$m];
    $average[$m] = $average[$index_of_min];
    $average[$index_of_min] = $temp;

    $temp = $names[$m];
    $names[$m] = $names[$index_of_min];
    $names[$index_of_min] = $temp;
}

echo $names[0], ", ", $names[1], ", ", $names[2], "\n";
?>

```

32. Solution

```

<?php
define("AREAS", 5);
define("HOURS", 48);

$names = [];
$CO2 = [[]];
for ($i = 0; $i <= AREAS - 1; $i++) {
    $names[$i] = trim(fgets(STDIN));
    for ($j = 0; $j <= HOURS - 1; $j++) {
        $CO2[$i][$j] = trim(fgets(STDIN));
    }
}

$average_per_hour = [];
for ($i = 0; $i <= AREAS - 1; $i++) {
    $average_per_hour[$i] = 0;
    for ($j = 0; $j <= HOURS - 1; $j++) {
        $average_per_hour[$i] += $CO2[$i][$j];
    }
    $average_per_hour[$i] /= HOURS;
}

for ($i = 0; $i <= AREAS - 1; $i++) {
    echo $names[$i], ", ", $average_per_hour[$i], "\n";
}

$average_per_city = [];
for ($j = 0; $j <= HOURS - 1; $j++) {
    $average_per_city[$j] = 0;
    for ($i = 0; $i <= AREAS - 1; $i++) {
        $average_per_city[$j] += $CO2[$i][$j];
    }
    $average_per_city[$j] /= AREAS;
}

for ($j = 0; $j <= HOURS - 1; $j++) {
    echo $average_per_city[$j], "\n";
}

$maximum = $average_per_city[0];
$m_j = 0;
for ($j = 1; $j <= HOURS - 1; $j++) {
    if ($average_per_city[$j] > $maximum) {
        $maximum = $average_per_city[$j];
        $m_j = $j;
    }
}
echo $m_j, "\n";

$maximum = $CO2[0][0];

```

```

$m_i = 0;
$m_j = 0;
for ($i = 0; $i <= AREAS - 1; $i++) {
    for ($j = 0; $j <= HOURS - 1; $j++) {
        if ($CO2[$i][$j] > $maximum) {
            $maximum = $CO2[$i][$j];
            $m_i = $i;
            $m_j = $j;
        }
    }
}
echo $m_j, ", ", $names[$m_i], "\n";

for ($m = 1; $m <= AREAS - 1; $m++) {
    $element_1 = $average_per_hour[$m];
    $element_2 = $names[$m];

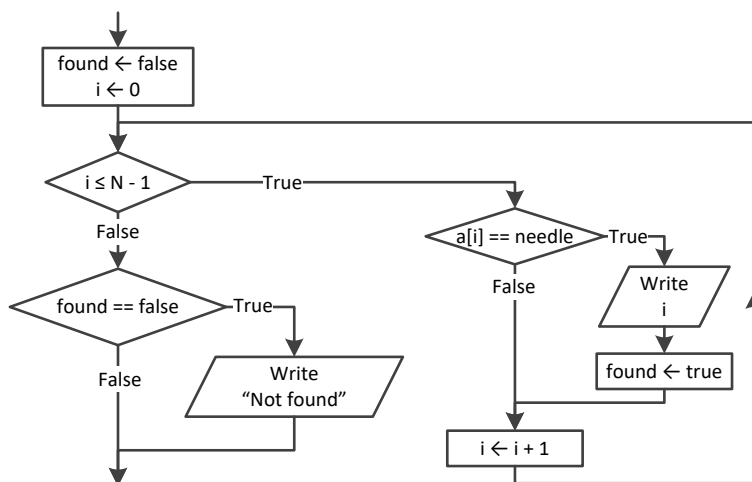
    $n = $m;
    while ($n > 0 && $average_per_hour[$n - 1] < $element_1) {
        $average_per_hour[$n] = $average_per_hour[$n - 1];
        $names[$n] = $names[$n - 1];
        $n--;
    }

    $average_per_hour[$n] = $element_1;
    $names[$n] = $element_2;
}

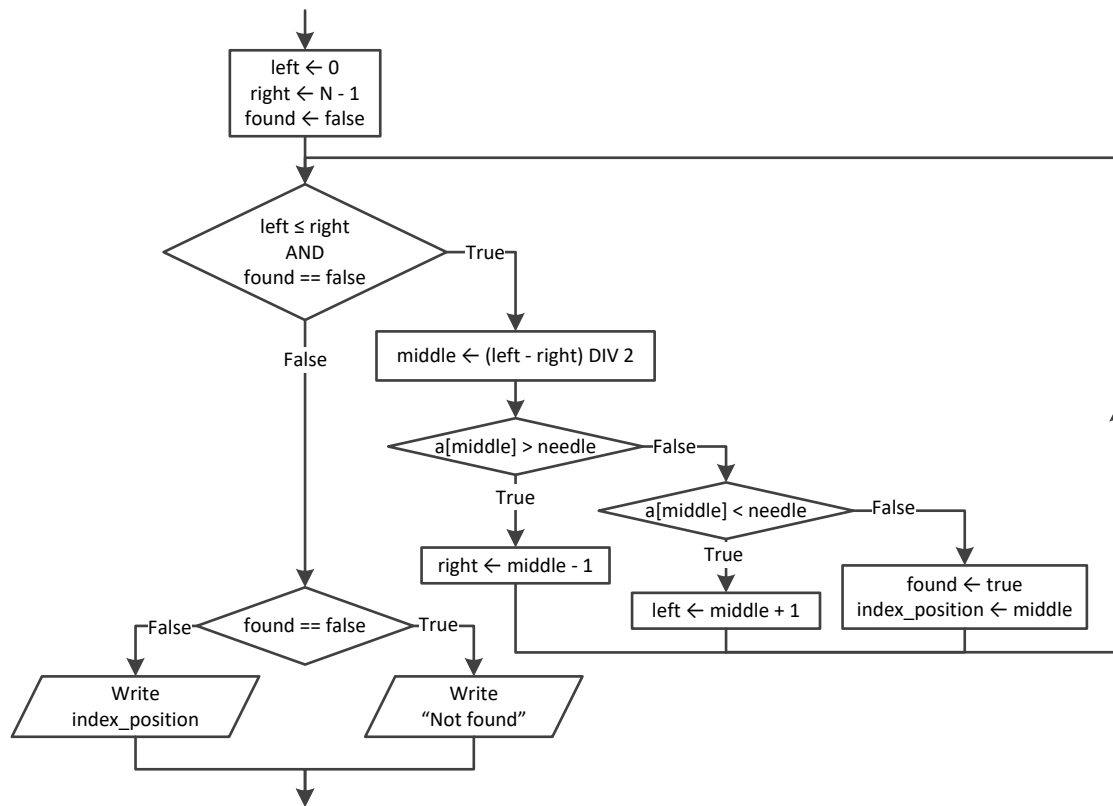
echo $names[0], ", ", $names[1], ", ", $names[2], "\n";
?>

```

33. Solution



34. Solution



35. Solution

```

<?php
define("TEAMS", 10);
define("GAMES", 16);

$names = [];
$goals_scored = [[]];
$goals_let_in = [[]];
for ($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][$j] = trim(fgets(STDIN));
        while ($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][$j] = trim(fgets(STDIN));
        while ($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\n";
}
else {
    $total = 0;
    for ($j = 0; $j <= GAMES - 1; $j++) {
        if ($goals_scored[$i][$j] > $goals_let_in[$i][$j]) {
            $total += 3;
        }
        elseif ($goals_scored[$i][$j] == $goals_let_in[$i][$j]) {
            $total += 1;
        }
    }
    echo $total, "\n";
}
?>

```

36. Solution

```

<?php
define("CLASS1", 20);
define("CLASS2", 25);

echo "Class 1\n";
$names1 = [];
for ($i = 0; $i <= CLASS1 - 1; $i++) {
    echo "Enter name: ";
    $names1[$i] = trim(fgets(STDIN));
}

echo "Class 2\n";
$names2 = [];
for ($i = 0; $i <= CLASS2 - 1; $i++) {
    echo "Enter name: ";
    $names2[$i] = trim(fgets(STDIN));
}

//Bubble sort
for ($m = 1; $m <= CLASS1 - 1; $m++) {
    for ($n = CLASS1 - 1; $n >= $m; $n--) {

```

```

        if ($names1[$n] < $names1[$n - 1]) {
            $temp = $names1[$n];
            $names1[$n] = $names1[$n - 1];
            $names1[$n] = $temp;
        }
    }
}
for ($m = 1; $m <= CLASS2 - 1; $m++) {
    for ($n = CLASS2 - 1; $n >= $m; $n--) {
        if ($names2[$n] < $names2[$n - 1]) {
            $temp = $names2[$n];
            $names2[$n] = $names2[$n - 1];
            $names2[$n] = $temp;
        }
    }
}

echo "\nClass 1\n";
for ($i = 0; $i <= CLASS1 - 1; $i++) {
    echo $names1[$i], "\n";
}
echo "\nClass 2\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) {
    $middle = (int)(($left + $right) / 2);

    if ($names1[$middle] < $needle) {
        $right = $middle - 1;
    }
    elseif ($names1[$middle] > $needle) {
        $left = $middle + 1;
    }
    else {
        $found = true;
    }
}

if ($found) {
    echo "Student found in Class No 1\n";
}
else {
    $left = 0;
    $right = CLASS2 - 1;

```

```

while ($left <= $right && !$found) {
    $middle = (int)(( $left + $right) / 2);

    if ($names2[$middle] < $needle) {
        $right = $middle - 1;
    }
    elseif ($names2[$middle] < $needle) {
        $left = $middle + 1;
    }
    else {
        $found = true;
    }
}

if ($found) {
    echo "Student found in Class No 2\n";
}
else {
    echo "Student not found in either class\n";
}
}
?>

```

37. Solution

```

echo "Enter username: ";
$usr = strtoupper(trim(fgets(STDIN)));
echo "Enter password: ";
$pwd = strtoupper(trim(fgets(STDIN)));

$i = 0;
while ($i < 99 && strtoupper($usernames[$i]) != $usr) {
    $i++;
}

if (strtoupper($usernames[$i]) == $usr && strtoupper($passwords[$i]) == $pwd) {
    echo "Login OK!\n";
}
else {
    echo "Login Failed!\n";
}

```

38. Solution

```

echo "Enter a value to search: ";
$value_str = trim(fgets(STDIN));

$found = false;

for ($i = 0; $i <= 999; $i++) {
    if ($names[$i] == $value_str) {
        echo $SSNs[$i], "\n";
        $found = true;
    }
}

```

```

    }
}

if (!$found) {
    $value = (int)$value_str;
    $i = 0;
    while ($i < 999 && $SSNs[$i] != $value) {
        $i++;
    }

    if ($SSNs[$i] == $value) {
        $found = true;
        echo $names[$i], "\n";
    }
}

if (!$found) {
    echo "This value does not exist\n";
}

```

39. Solution

```

<?php
define("STUDENTS", 12);
define("LESSONS", 6);

$grades = [[]];
for ($i = 0; $i <= STUDENTS - 1; $i++) {
    for ($j = 0; $j <= LESSONS - 1; $j++) {
        do {
            $grades[$i][$j] = trim(fgets(STDIN));
            $failure = false;
            if ($grades[$i][$j] < 0) {
                echo "Error! You entered a negative value\n";
                $failure = true;
            }
            elseif ($grades[$i][$j] > 100) {
                echo "Error! You entered a value grater than 100\n";
                $failure = true;
            }
        } while ($failure);
    }
}

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

```

```

$found = false;
for ($i = 0; $i <= STUDENTS - 1; $i++) {
    if ($average[$i] < 70) {
        $found = true;
        break;
    }
}

if ($found) {
    echo "There is at least one student that has an average value below 70\n";
}
?>

```

40. Solution

```

<?php
$morseAlphabet = [
    "A" => ".-",
    "B" => "-...",
    "C" => "-.-.",
    "D" => "-..",
    "E" => ".",
    "F" => "...-",
    "G" => "--.",
    "H" => "....",
    "I" => "..",
    "J" => ".---",
    "K" => "-.-",
    "L" => ".-..",
    "M" => "--",
    "N" => "-.",
    "O" => "---",
    "P" => ".--.",
    "Q" => "--.-",
    "R" => "-.-.",
    "S" => "...",
    "T" => "-",
    "U" => "-.-",
    "V" => "...-",
    "W" => ".--",
    "X" => "-.-.-",
    "Y" => "-.-.-",
    "Z" => "--..",
    " ", "/"
];

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

for ($i = 0; $i <= strlen($word) - 1; $i++) {
    $letter = $word[$i];
    echo $morseAlphabet[strtoupper($letter)], " ";
}

```

```
}  
?>
```

Review in “Arrays in PHP”

Review Crossword Puzzle

1.



35.4 Review Questions: True/False

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

7. true
8. false
9. true
10. true
11. false
12. true

36.8 Review Questions: True/False

- Chapter 36
1. false

2. true

3. false

4. true

5. true

6. false

7. true

8. false

9. true

10. false

11. true

12. true

13. true

14. true

15. true

16. false

17. false

18. true

19. false

20. true

21. true

22. true

23. true

24. false

25. true

26. false

27. true

28. false

29. true

30. true

31. true

32. true

33. false

36.9 Review Exercises

1. Solution

```
function find_max($a, $b) {
  if ($a > $b) {
    $maximum = $a;
  }
  else {
    $maximum = $b;
  }
  return $maximum;
}
```

2. Solution

Step	Statement	Main Code		Method sum_digits()		
		\$s	\$i	\$a	\$d1	\$d2
1	\$s = 0	0	?			
2	\$i = 25	0	25			
3	\$i <= 27	true				
4	\$s += sum_digits(\$i)			25	?	?
5	\$d1 = \$a % 10			25	5	?
6	\$d2 = (int)(\$a / 10)			25	5	2

7	return \$d1 + \$d2	7	25			
8	\$i++	7	26			
9	\$i <= 27	true				
10	\$s += sum_digits(\$i)			26	?	?
11	\$d1 = \$a % 10			26	6	?
12	\$d2 = (int)(\$a / 10)			26	6	2
13	return \$d1 + \$d2	15	26			
14	\$i++	15	27			
15	\$i <= 27	true				
16	\$s += sum_digits(\$i)			27	?	?
17	\$d1 = \$a % 10			27	7	?
18	\$d2 = (int)(\$a / 10)			27	7	2
19	return \$d1 + \$d2	24	27			
20	\$i++	24	28			
21	\$i <= 27	false				
22	echo \$s	It displays: 24				

3. Solution

Step	Statement	Main Code		Method sss()		
		\$s	\$i	\$a	\$total	\$k
1	\$i = 1	?	1			
2	\$s = 0	0	1			
3	while(\$i < 6)	true				
4	if (\$i % 2 == 1)	true				
5	\$s += 1	1	1			
6	\$i++	1	2			
7	while(\$i < 6)	true				
8	if (\$i % 2 == 1)	false				
9	\$s += sss(\$i)			2	?	?
10	\$total = 0			2	0	?
11	\$k = 1			2	0	1
12	\$k <= \$a			true		
13	\$total += \$k			2	1	1
14	\$k++			2	1	2
15	\$k <= \$a			true		
16	\$total += \$k			2	3	2
17	\$k++			2	3	3

18	<code>\$k <= \$a</code>			false		
19	<code>return \$total</code>	4	2			
20	<code>\$i++</code>	4	3			
21	<code>while(\$i < 6)</code>	true				
22	<code>if (\$i % 2 == 1)</code>	true				
23	<code>\$s += 1</code>	5	3			
24	<code>\$i++</code>	5	4			
25	<code>while(\$i < 6)</code>	true				
26	<code>if (\$i % 2 == 1)</code>	false				
27	<code>\$s += sss(\$i)</code>			4	?	?
28	<code>\$total = 0</code>			4	0	?
29	<code>\$k = 1</code>			4	0	1
30	<code>\$k <= \$a</code>			true		
31	<code>\$total += \$k</code>			4	1	1
32	<code>\$k++</code>			4	1	2
33	<code>\$k <= \$a</code>			true		
34	<code>\$total += \$k</code>			4	3	2
35	<code>\$k++</code>			4	3	3
36	<code>\$k <= \$a</code>			true		
37	<code>\$total += \$k</code>			4	6	4
38	<code>\$k++</code>			4	6	4
39	<code>\$k <= \$a</code>			true		
40	<code>\$total += \$k</code>			4	10	4
41	<code>\$k++</code>			4	10	5
42	<code>\$k <= \$a</code>			false		
43	<code>return \$total</code>	15	4			
44	<code>\$i++</code>	15	5			
45	<code>while(\$i < 6)</code>	true				
46	<code>if (\$i % 2 == 1)</code>	true				
47	<code>\$s += 1</code>	16	5			
48	<code>\$i++</code>	16	6			
49	<code>while(\$i < 6)</code>	false				
50	<code>echo \$s</code>	It displays: 16				

4. Solution

Step	Statement	Main Code				Method 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 (int)((\$b + \$d) / 2)	12	2	1	4		
8	echo \$m, " ", \$a, " ", \$x;	It displays: 2 1 4					
9	\$a += 2	12	2	3	4		
10	\$m++	12	3	3	4		
11	while (\$a < 6)	true					
12	if (\$k % \$m != 0)	false					
13	\$x = \$a + \$m + custom_div(\$m, \$a)					3	3
14	return (int)((\$b + \$d) / 2)	12	3	3	9		
15	echo \$m, " ", \$a, " ", \$x;	It displays: 3 3 9					
16	\$a += 2	12	3	5	9		
17	\$m++	12	4	5	9		
18	while (\$a < 6)	true					
19	if (\$k % \$m != 0)	false					
20	\$x = \$a + \$m + custom_div(\$m, \$a)					4	5
21	return (int)((\$b + \$d) / 2)	12	4	5	13		
22	echo \$m, " ", \$a, " ", \$x;	It displays: 4 5 13					
23	\$a += 2	12	4	7	13		
24	\$m++	12	5	7	13		
25	while (\$a < 6)	false					

5. Solution

Step	Statement	Main Code		void Method display()
		\$i	\$x	\$a
1	\$i = 1	1	?	
2	\$i <= 5	true		
3	\$x = trim(fgets(STDIN))	1	3	

4	display(\$x)			3
5	if (\$a % 2 == 0)			false
6	echo \$a, " is odd"	It displays: 3 is odd		
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"	It displays: 7 is odd		
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"	It displays: 9 is odd		
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"	It displays: 2 is even		
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"	It displays: 4 is even		
31	\$i++	6	4	
32	\$i <= 5	false		

6. Solution

Step	Statement	Main Code		void Method division()	
		\$x	\$y	\$a	\$b
1	\$x = 20	20	?		
2	\$y = 30	20	30		
3	while (\$x % \$y < 30)	true			

4	<code>division(\$y, \$x)</code>			30	20
5	<code>\$b = (int)(\$b / \$a)</code>			30	0
6	<code>echo \$a * \$b</code>	It displays: 0			
7	<code>\$x = 4 * \$y</code>	120	30		
8	<code>\$y++</code>	120	31		
9	<code>while (\$x % \$y < 30)</code>	true			
10	<code>division(\$y, \$x)</code>			31	120
11	<code>\$b = (int)(\$b / \$a)</code>			31	3
12	<code>echo \$a * \$b</code>	It displays: 93			
13	<code>\$x = 4 * \$y</code>	124	31		
14	<code>\$y++</code>	124	32		
15	<code>while (\$x % \$y < 30)</code>	true			
16	<code>division(\$y, \$x)</code>			32	124
17	<code>\$b = (int)(\$b / \$a)</code>			32	3
18	<code>echo \$a * \$b</code>	It displays: 96			
19	<code>\$x = 4 * \$y</code>	128	32		
20	<code>\$y++</code>	128	33		
21	<code>while (\$x % \$y < 30)</code>	true			
22	<code>division(\$y, \$x)</code>			33	128
23	<code>\$b = (int)(\$b / \$a)</code>			33	3
24	<code>echo \$a * \$b</code>	It displays: 99			
25	<code>\$x = 4 * \$y</code>	132	33		
26	<code>\$y++</code>	132	34		
27	<code>while (\$x % \$y < 30)</code>	false			

7. Solution

Step	Statement	Main Code		void Method calculate()		
		\$i	\$m	\$n	\$s	\$j
1	<code>\$i = 1</code>	1	?			
2	<code>\$i <= 3</code>	true				
3	<code>\$m = trim(fgets(STDIN))</code>	1	2			
4	<code>calculate(\$m)</code>			2	?	?
5	<code>\$s = 0</code>			2	0	?
6	<code>\$j = 2</code>			2	0	2
7	<code>\$j <= 2 * \$n</code>			true		
8	<code>\$s = \$s + \$j ** 2</code>			2	4	2

9	\$j += 2			2	4	4
10	\$j <= 2 * \$n			true		
11	\$s = \$s + \$j ** 2			2	20	4
12	\$j += 2			2	20	6
13	\$j <= 2 * \$n			false		
14	\$m = trim(fgets(STDIN))	It displays: 20				
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 + \$j ** 2			3	4	2
23	\$j += 2			3	4	4
24	\$j <= 2 * \$n			true		
25	\$s = \$s + \$j ** 2			3	20	4
26	\$j += 2			3	20	6
27	\$j <= 2 * \$n			true		
28	\$s = \$s + \$j ** 2			3	56	6
29	\$j += 2			3	56	8
30	\$j <= 2 * \$n			false		
31	echo \$s	It displays: 56				
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 + \$j ** 2			4	4	2
40	\$j += 2			4	4	4
41	\$j <= 2 * \$n			true		
42	\$s = \$s + \$j ** 2			4	20	4
43	\$j += 2			4	20	6
44	\$j <= 2 * \$n			true		
45	\$s = \$s + \$j ** 2			4	56	6

46	\$j += 2			4	56	8
47	\$j <= 2 * \$n			true		
48	\$s = \$s + \$j ** 2			4	120	8
49	\$j += 2			4	120	10
50	\$j <= 2 * \$n			false		
51	echo \$s	It displays: 120				
52	\$i++	4	4			
53	\$i <= 3	false				

8. Solution

```
function find_sum($a, $b, $c) {
    return $a + $b + $c;
}
```

9. Solution

```
function find_avg($a, $b, $c, $d) {
    return ($a + $b + $c + $d) / 4;
}
```

10. Solution

```
function maximum($a, $b, $c) {
    $m = $a;
    if ($b > $m) {
        $m = $b;
    }
    if ($c > $m) {
        $m = $c;
    }
    return $m;
}
```

11. Solution

```
function display_max($a, $b, $c, $d, $e) {
    $m = $a;
    if ($b > $m) {
        $m = $b;
    }
    if ($c > $m) {
        $m = $c;
    }
    if ($d > $m) {
        $m = $d;
    }
    if ($e > $m) {
        $m = $e;
    }
}
```



```

    }
    echo $m;
}

```

12. Solution

```

function my_round($x) {
    $digit_to_check = (int)($x * 1000) % 10;
    if ($digit_to_check >= 5) {
        $return_value = ((int)($x * 100) + 1) / 100.0;
    }
    else {
        $return_value = ((int)($x * 100)) / 100.0;
    }

    return $return_value;
}

```

13. Solution

```

<?php
function find_min($a, $b) {
    $minimum = $a;
    if ($b < $minimum) {
        $minimum = $b;
    }
    return $minimum;
}

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), "\n";

//Second approach
echo find_min(find_min($x1, $x2), find_min($x3, $x4)), "\n";
?>

```

14. 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), "\n";
echo "Celsius: ", Kelvin_to_Celsius($k), "\n";
?>

```

15. Solution

```

<?php
function bmi($w, $h) {
    $b = $w * 703 / $h ** 2;
    if ($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): ";
$weight = trim(fgets(STDIN));
while ($weight < 0) {
    echo "Error! Enter your weight (in pounds): ";
    $weight = trim(fgets(STDIN));
}

echo "Enter your age: \n";
$age = trim(fgets(STDIN));
while ($age < 18) {
    echo "Error! Enter your age: ";
    $age = trim(fgets(STDIN));
}

echo "Enter your height (in inches): \n";
$height = trim(fgets(STDIN));
while ($height < 0) {
    echo "Error! Enter your height (in inches): \n";
    $height = trim(fgets(STDIN));
}

```

```
    echo bmi($weight, $height), "\n";  
?>
```

16. 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);  
}  
?>
```

17. 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\n";  
}  
  
function miles_to_meters($miles) {
```

```

    echo $miles, " miles equals ", ($miles * 1609.344), " meters\n";
}

display_menu();
$choice = trim(fgets(STDIN));
while ($choice != 3) {
    echo "Enter distance: \n";
    $distance = trim(fgets(STDIN));
    if ($choice == 1) {
        meters_to_miles($distance);
    }
    else {
        miles_to_meters($distance);
    }

    display_menu();
    $choice = trim(fgets(STDIN));
}
?>

```

18. 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, "\n";
}

echo "Enter number of seconds: ";
$seconds = trim(fgets(STDIN));
amount_to_pay($seconds);
?>

```

Chapter 37

37.9 Review Questions: True/False

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

37.10 Review Exercises**1. Solution**

It displays: 5

2. Solution

It displays: 14

3. Solution

It displays: 14

4. Solution

Step	Statement	Main Code				void Method 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)	1	1	true				
5	if (\$k % \$m != 0)	1	1	false				
6	\$x = \$a + \$m + (int)(\$a - \$m)	1	1	12	2			
7	echo \$m, " ", \$a, " ", \$x	It displays: 1 1 2						
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	<code>\$x = \$y</code>					2	2	3
13	<code>\$y = \$temp</code>					2	3	3
14	<code>while (\$a < 8)</code>	2	3	12	2			
		2	3	true				
15	<code>if (\$k % \$m != 0)</code>	2	3	false				
16	<code>\$x = \$a + \$m + (int)(\$a - \$m)</code>	2	3	12	4			
17	<code>echo \$m, " ", \$a, " ", \$x</code>	It displays: 3 2 4						
18	<code>\$a += 2</code>	4	3	12	4			
19	<code>\$m++</code>	4	4	12	4			
20	<code>swap(\$a, \$m)</code>					4	4	?
21	<code>\$temp = \$x</code>					4	4	4
22	<code>\$x = \$y</code>					4	4	4
23	<code>\$y = \$temp</code>					4	4	4
24	<code>while (\$a < 8)</code>	4	4	12	4			
		4	4	true				
25	<code>if (\$k % \$m != 0)</code>	4	4	false				
26	<code>\$x = \$a + \$m + (int)(\$a - \$m)</code>	4	4	12	8			
27	<code>echo \$m, " ", \$a, " ", \$x</code>	It displays: 4 4 8						
28	<code>\$a += 2</code>	6	4	12	8			
29	<code>\$m++</code>	6	5	12	8			
30	<code>swap(\$a, \$m)</code>					6	5	?
31	<code>\$temp = \$x</code>					6	5	6
32	<code>\$x = \$y</code>					5	5	6
33	<code>\$y = \$temp</code>					5	6	5
34	<code>while (\$a < 8)</code>	5	6	12	8			
		5	6	true				
35	<code>if (\$k % \$m != 0)</code>	5	6	false				
36	<code>\$x = \$a + \$m + (int)(\$a - \$m)</code>	5	6	12	10			
37	<code>echo \$m, " ", \$a, " ", \$x</code>	It displays: 6 5 10						
38	<code>\$a += 2</code>	7	6	12	10			
39	<code>\$m++</code>	7	7	12	10			
40	<code>swap(\$a, \$m)</code>					7	7	?
41	<code>\$temp = \$x</code>					7	7	7
42	<code>\$x = \$y</code>					7	7	7
43	<code>\$y = \$temp</code>					7	7	7
44	<code>while (\$a < 8)</code>	7	7	12	10			
		7	7	true				

45	if (\$k % \$m != 0)	7	7	true			
46	\$x = \$a % \$m	7	7	12	0		
47	swap(\$a, \$m)					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	7	7	12	0		
	It displays: 7 7 0						
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

It displays: hellohellohello

6. Solution

It displays: 15

7. Solution

It displays: 11 4

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) {
```

```

    $average = [];
    for ($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 = [];
$grades = [[]];

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 = strtolower(trim(fgets(STDIN)));
    return $message;
}

function part2($message) {
    $message_clean = "";
    for ($i = 0; $i <= strlen($message) - 1; $i++) {
        $letter = $message[$i];
        if ($letter != " " && $letter != "," && $letter != "." && $letter != "?") {
            $message_clean .= $letter;
        }
    }
    return $message_clean;
}

function part3($message_clean) {
    $middle_pos = (int)((strlen($message_clean) - 1) / 2);
    $j = strlen($message_clean) - 1;
    $palindrome = true;
    for ($i = 0; $i <= $middle_pos; $i++) {
        $left_letter = $message_clean[$i];
        $right_letter = $message_clean[$j];
        if ($left_letter != $right_letter) {
            $palindrome = false;
            break;
        }
        $j--;
    }
    return $palindrome;
}

function part4($message) {
    $message_clean = part2($message);
    $palindrome = part3($message_clean);
    return $palindrome;
}

$message = part1();
$palindrome = part4($message);
if ($palindrome) {
    echo "The message is palindrome\n";
}
?>

```

10. Solution

```

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

```

```

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

echo $maximum;
?>

```

11. Solution

```

function f1($a, $b, $c, &$returning_array) {
    $returning_array[0] = $a + $b + $c;
    $returning_array[1] = $returning_array[0] / 3;
}

```

12. Solution

```

function my_round($x, $decimal_places = 2) {
    $digit_to_check = (int)($x * (10 ** ($decimal_places + 1))) % 10;
    if ($digit_to_check >= 5) {
        $return_value = (int)(($x * 10 ** $decimal_places) + 1) / 10 ** $decimal_places;
    }
    else {
        $return_value = (int)($x * 10 ** $decimal_places) / 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 == "YES"; //This returns true or false
}

function find_area($b, $h) {
    return $b * $h;
}

do {
    echo "Enter the base of the parallelogram: ";
}

```

```

$b = trim(fgets(STDIN));
echo "Enter the height of the parallelogram: ";
$h = trim(fgets(STDIN));

echo "Area = ", find_area($b, $h), "\n";

echo "Would you like to repeat? \n";
} while (get_input());
?>

```

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

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], "\n";
    }
}
?>

```

15. Solution

```

<?php
define("JUDGES", 10);

function get_array() {
    $score = [];

    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($score, &$amp;minimum, &$amp;maximum) {
    $minimum = $score[0];
    $maximum = $score[0];
    for ($i = 1; $i <= JUDGES - 1; $i++) {
        if ($score[$i] > $maximum) {
            $maximum = $score[$i];
        }
        if ($score[$i] < $minimum) {
            $minimum = $score[$i];
        }
    }
}

echo "Enter artist's name: ";
$name = trim(fgets(STDIN));
$score = get_array();
find_min_max($score, $minimum, $maximum);

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

$points = $total - $minimum - $maximum;
echo "Artist ", $name, " got ", $points, " points\n";
?>

```

16. Solution

```

<?php

```

```

function woc($index) {
    if ($index == 1) {
        $return_value = 1;
    }
    else {
        $return_value = 2 * woc($index - 1);
    }
    return $return_value;
}

$total = 0;
for ($i = 1; $i <= 64; $i++) {
    $total += woc($i);
}
echo $total, "\n";
?>

```

17. Solution

```

<?php
function factorial($value) {
    if ($value == 1) {
        $return_value = 1;
    }
    else {
        $return_value = $value * factorial($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) + $x ** $i / factorial($i) - $x ** ($i - 2) / factorial($i - 2);
    }

    return $return_value;
}

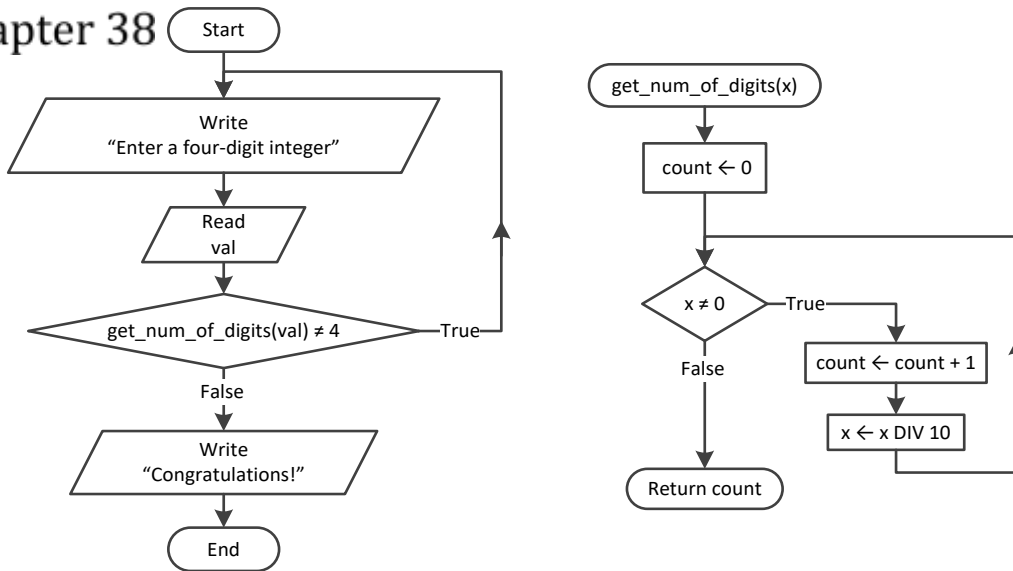
echo my_cos(pi() / 4), "\n";
?>

```

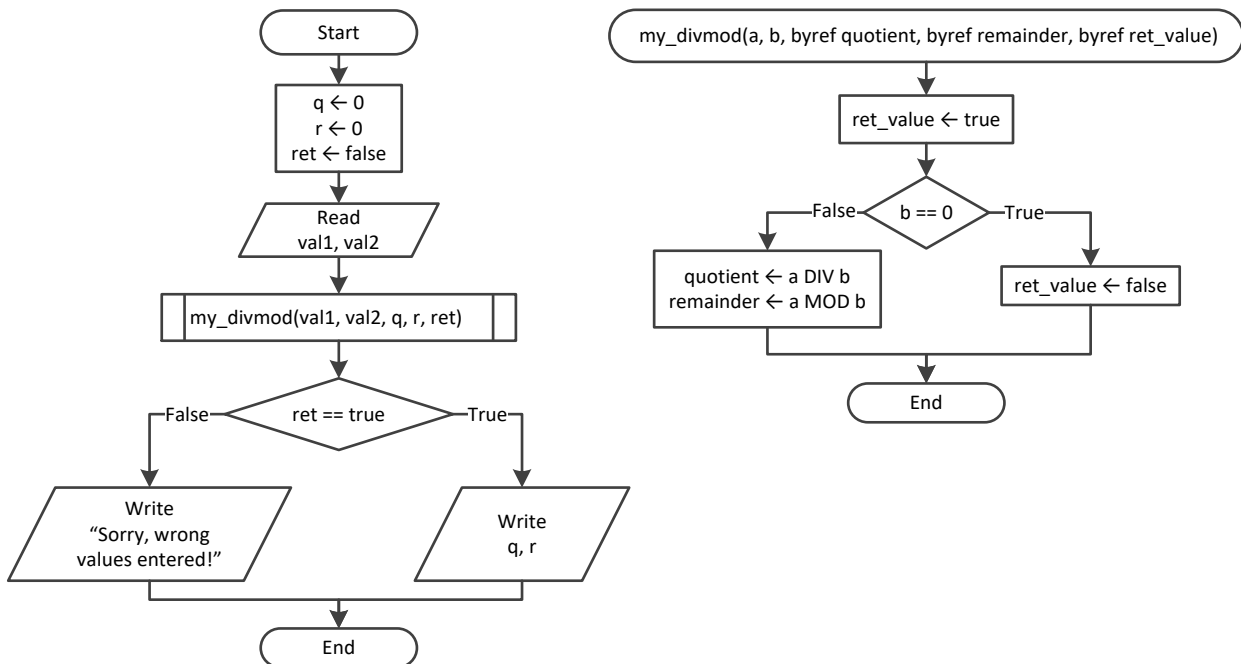
38.3 Review Exercises

1. Solution

Chapter 38



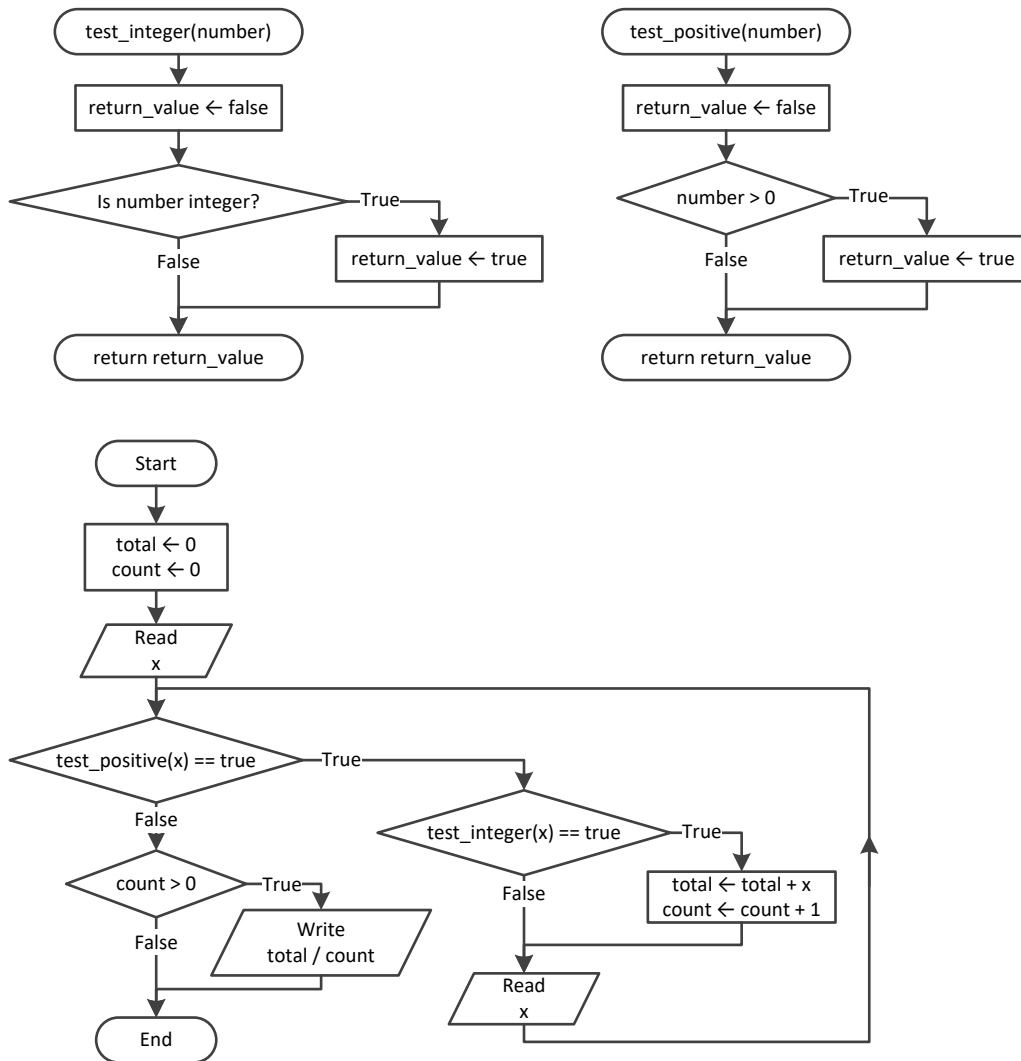
2. Solution



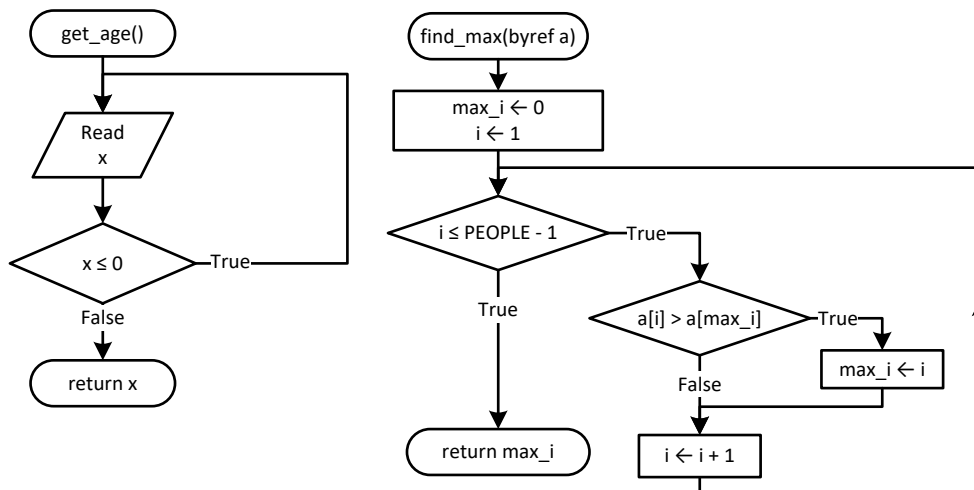
Flowcharts are a loose method of representing an algorithm. Thus, you can represent a pass by reference using the keyword `byref`, which clearly denotes what it actually does.

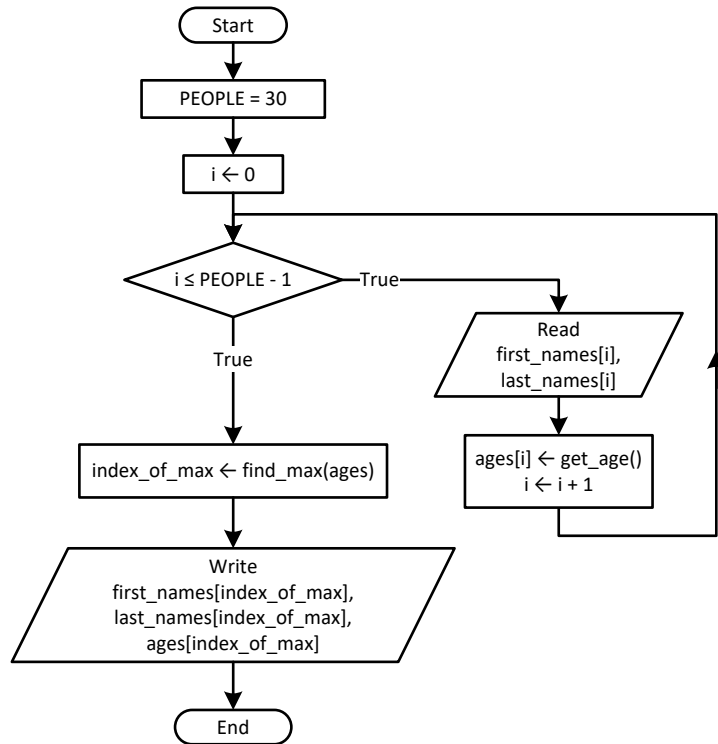
Some programmers, instead of using the keyword `byref`, prefer to write the keyword `inout`, which denotes pretty much the same thing—that the variable is both input (it accepts values) and output (it returns values).

3. Solution

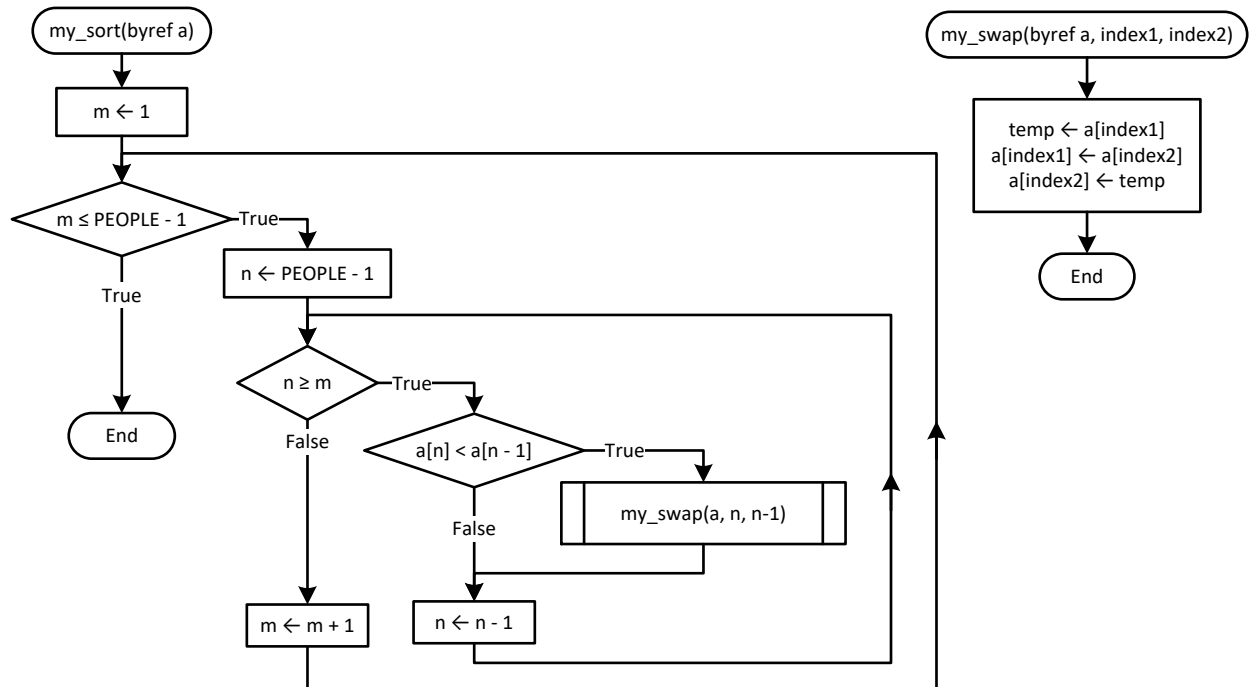


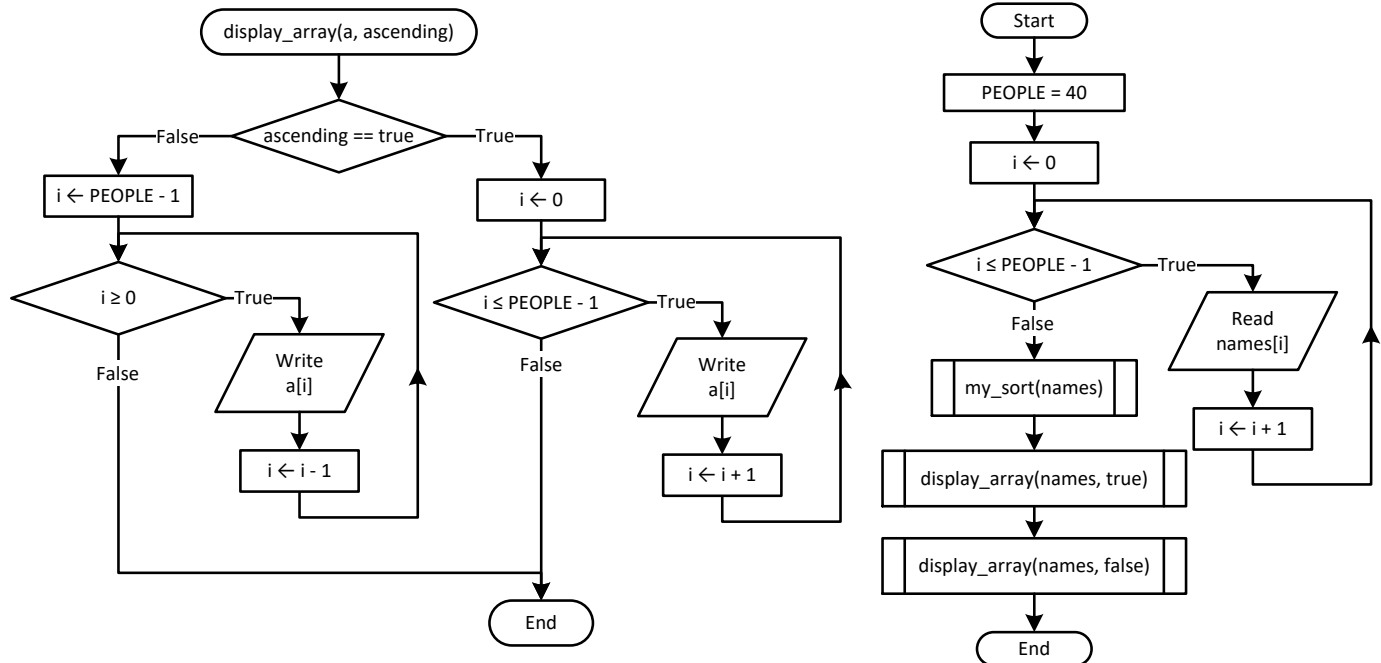
4. Solution





5. Solution





6. Solution

```

<?php
function f1($n) {
    $s = 0;
    for ($i = 1; $i <= $n; $i++) {
        if ($i < $n / 2.0) {
            $s += $n ** 2;
        }
        else {
            $s += $n ** 3;
        }
    }
    return $s;
}

do {
    echo "Enter a positive integer ";
    $val = trim(fgets(STDIN));
} while ($val < 0);
echo f1($val), "\n";
?>

```

7. Solution

```

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

function read_values() {
    $values = [];

```

```


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

function find_min_max($values, &$min_i, &$max_i) {
    $min_i = $max_i = 0;
    for ($i = 1; $i <= ELEMENTS - 1; $i++) {
        if ($values[$i] < $values[$min_i]) {
            $min_i = $i;
        }
        if ($values[$i] > $values[$max_i]) {
            $max_i = $i;
        }
    }
}

$v = [];
$min_i = 0;
$max_i = 0;

$v = read_values();
find_min_max($v, $min_i, $max_i);
echo $v[$min_i], ", ", $v[$max_i], "\n";
?>

```

 Please note the way the void method `find_min_max()` finds the index positions of the minimum and the maximum values of the array values. This method is not the same as the one you learned in paragraph 34.3; however, it can be used as an alternative.

8. Solution

```

<?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 * $x ** $i / factorial($i);

        $sign = -$sign;
    } while ($i < 10);
    return $sinus;
}

```

```

    $i += 2;
} while (abs($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";
}
?>

```

9. 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)) {
                $days = 29;
            }
            else {
                $days = 28;
            }
            break;
        default:
            $days = 31;
    }

    return $days;
}

function check_date($day, $month, $year) {
    $return_value = true;
    if ($month < 1 || $month > 12) {
        $return_value = false;
    }
}

```

```

    }
    elseif ($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)) {
    echo "Error!\n";
    echo "Enter day: ";
    $day = trim(fgets(STDIN));
    echo "Enter month: ";
    $month = trim(fgets(STDIN));
    echo "Enter year: ";
    $year = trim(fgets(STDIN));
}

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

echo $total, "\n";
?>

```

10. Solution

```

<?php
function display_menu() {
    echo "-----\n";
    echo "1. Convert USD to Euro (EUR)\n";
    echo "2. Convert USD to British Pound Sterling (GBP)\n";
    echo "3. Convert EUR to USD\n";
    echo "4. Convert EUR to GBP\n";
    echo "5. Convert GBP to USD\n";
    echo "6. Convert GBP to EUR\n";
    echo "7. Exit\n";
    echo "-----\n";
    echo "Enter a choice: ";
}

function USD_to_EUR($value) {
    return $value * 0.87;
}

```

```

function USD_to_GBP($value) {
    return $value * 0.76;
}

display_menu();
$choice = trim(fgets(STDIN));
while ($choice != 7) {
    echo "Enter an amount: ";
    $amount = trim(fgets(STDIN));
    switch ($choice) {
        case 1:
            echo $amount, " USD = ", USD_to_EUR($amount), " Euro\n";
            break;
        case 2:
            echo $amount, " USD = ", USD_to_GBP($amount), " GBP\n";
            break;
        case 3:
            echo $amount, " EUR = ", 1 / USD_to_EUR(1 / $amount), " USD\n";
            break;
        case 4:
            echo $amount, " EUR = ", USD_to_GBP(1 / USD_to_EUR(1 / $amount)), " GBP\n";
            break;
        case 5:
            echo $amount, " GBP = ", 1 / USD_to_GBP(1 / $amount), " USD\n";
            break;
        case 6:
            echo $amount, " GBP = ", USD_to_EUR(1 / USD_to_GBP(1 / $amount)), " EUR\n";
            break;
    }

    display_menu();
    $choice = trim(fgets(STDIN));
}
?>

```

11. Solution

```

<?php
function dice() {
    return rand(1, 6);
}

$names = [];

echo "Player1 - Enter name: ";
$names[0] = trim(fgets(STDIN));
echo "Player2 - Enter name: ";
$names[1] = trim(fgets(STDIN));

for ($player = 0; $player <= 1; $player++) {
    $total = 0;
    for ($i = 1; $i <= 10; $i++) {

```

```

    echo $names[$player], ", hit enter to roll the dice!\n";
    fgets(STDIN); //This statement just waits the user to hit the enter key

    $dice1 = dice();
    $dice2 = dice();
    echo $dice1, " ", $dice2, "\n";
    $total += $dice1 + $dice2;
}
if ($player == 1) {
    $total_player1 = $total;
}
else {
    $total_player2 = $total;
}
}

if ($total_player1 == $total_player2) {
    echo "Tie!\n";
}
elseif ($total_player1 > $total_player2) {
    echo $names[0], " wins\n";
}
else {
    echo $names[1], " wins\n";
}
}
?>

```

12. 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: ";
    return trim(fgets(STDIN));
}

function get_days() {
    echo "Enter total number of rental days: ";
    return trim(fgets(STDIN));
}

function get_charge($car_type, $rental_days) {
    if ($car_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 ($car_type == DIESEL) {
    if ($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 = [];
$rented_days = [];

for ($i = 0; $i <= CARS - 1; $i++) {
    $rented_car_types[$i] = get_choice();
    $rented_days[$i] = get_days();
}

$total = 0;
for ($i = 0; $i <= CARS - 1; $i++) {
    $charge = get_charge($rented_car_types[$i], $rented_days[$i]);
    echo "Car No ", ($i + 1), ": ", $charge, "\n";
    $total += $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: ", $total / (1 + TAX_RATE), "\n";
?>

```

13. Solution

```

<?php
define("CHANNELS", 10);
define("DAYS", 7);
define("DAY_NAMES", ["Monday", "Tuesday", "Wednesday", "Thursday", "Friday", "Saturday", "Sunday"]);

function get_data(&$names, &$viewers) {
    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) {
    $total = 0;
    for ($i = 0; $i <= 4; $i++) {
        $total += $a[$i];
    }
    return $total / 5.0;
}

$names = [];
$viewers = [[]];
get_data($names, $viewers);

$temporary_array = [];
for ($i = 0; $i <= CHANNELS - 1; $i++) {
    for ($j = 0; $j <= 4; $j++) {
        $temporary_array[$j] = $viewers[$i][$j];
    }
    $weekend = ($viewers[$i][DAYS - 2] + $viewers[$i][DAYS - 1]) / 2;
    if ($weekend >= 1.2 * get_average($temporary_array)) {
        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) {
        echo $names[$i], "\n";
    }
}
?>

```

14. Solution

```

<?php
define("CITIZENS", 300);

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++) {
        $minimum = $SSNs[$m];
        $index_of_min = $m;
        for ($n = $m; $n <= CITIZENS - 1; $n++) {
            if ($SSNs[$n] < $minimum) {
                $minimum = $SSNs[$n];
                $index_of_min = $n;
            }
        }
        $temp = $SSNs[$m];
        $SSNs[$m] = $SSNs[$index_of_min];
        $SSNs[$index_of_min] = $temp;
        $temp = $answers[$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) {
        $middle = (int)(( $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) {
    echo "SSN not found!\n";
    $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;
}

$SSNs = [];
$answers = [];

do {
    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, "\n";
    }
    echo "Repeat? \n";
    $answer = trim(fgets(STDIN));
} while ($answer == "yes");
?>

```

15. 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) {
            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 = [];  
$results = [[]];  
  
input_data($names, $results);  
display_result($names, $results);  
  
$index = find_team($names);  
while ($index != -1) {  
    $total = 0;  
    for ($j = 0; $j <= GAMES - 1; $j++) {  
        if ($results[$index][$j] == "W") {  
            $total += 3;  
        }  
        elseif ($results[$index][$j] == "T") {  
            $total += 1;  
        }  
    }  
    echo "Points: ", $total, "\n";  
    $index = find_team($names);  
}  
  
if ($index == -1) {  
    echo "Team not found\n";  
}  
?  
>
```

Review in “Subprograms”

Review Crossword Puzzle

1.



39.9 Review Questions: True/False

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

Chapter 39

39.10 Review Exercises

1. Solution

```
<?php
class Trigonometry {
    function square_area($side) {
        return $side * $side;
    }

    function rectangle_area($b, $h) {
        return $b * $h;
    }

    function triangle_area($b, $h) {
        return $b * $h / 2;
    }
}

$str = new Trigonometry();

echo "Enter square side: ";
$sqr_side = trim(fgets(STDIN));

echo "Enter rectangle base: ";
$rctngl_base = trim(fgets(STDIN));
echo "Enter rectangle height: ";
$rctngl_height = trim(fgets(STDIN));

echo "Enter triangle base: ";
$trngl_base = trim(fgets(STDIN));
echo "Enter triangle height: ";
$trngl_height = trim(fgets(STDIN));

echo $str->square_area($sqr_side), "\n";
echo $str->rectangle_area($rctngl_base, $rctngl_height), "\n";
echo $str->triangle_area($trngl_base, $trngl_height), "\n";
?>
```

2. Solution

```
<?php
```

```

class Pet {
    public $kind;
    public $legs_number;

    function start_running() {
        echo "Pet is running\n";
    }

    function stop_running() {
        echo "Pet stopped\n";
    }
}

$pet1 = new Pet();
$pet1->kind = "dog";
$pet1->legs_number = 4;

$pet2 = new Pet();
$pet2->kind = "monkey";
$pet2->legs_number = 2;

$pet1->start_running();
$pet2->start_running();
$pet1->stop_running();
?>

```

3. Solution

```

<?php
class Pet {
    private $_kind;
    private $_legs_number;

    //Define the constructor
    function __construct($kind, $legs_number) {
        $this->setKind($kind);
        $this->setLegs_number($legs_number);
    }

    function getKind() {
        return $this->_kind;
    }

    function setKind($value) {
        if ($value != "") {
            $this->_kind = $value;
        }
        else {
            throw new Exception("Cannot be empty");
        }
    }

    function getLegs_number() {

```

```

        return $this->_legs_number;
    }

    function setLegs_number($value) {
        if ($value >= 0) {
            $this->_legs_number = $value;
        }
        else {
            throw new Exception("Cannot be negative");
        }
    }

    function start_running() {
        echo "Pet is running\n";
    }

    function stop_running() {
        echo "Pet stopped\n";
    }
}

$pet1 = new Pet("dog", 4);

$pet1->start_running();
$pet1->stop_running();

$pet1->setKind(""); //This will throw an error
$pet1->setLegs_number(-1); //This will throw an error
?>

```

4. Solution

```

<?php
class Box {
    private $_width;
    private $_length;
    private $_height;

    //Define the constructor
    function __construct ($w, $l, $h) {
        //Initialize fields
        $this->_width = $w;
        $this->_length = $l;
        $this->_height = $h;
    }

    function display_volume() {
        echo "Volume: ", ($this->_width * $this->_length * $this->_height), "\n";
    }

    function display_dimensions() {
        echo $this->_width, " x ", $this->_length, " x ", $this->_height, "\n";
    }
}

```



```

}

define("BOXES", 3);

$list_of_obj = []; //create an array

for ($i = 0; $i <= BOXES - 1; $i++) {
    echo "Enter width: ";
    $w = trim(fgets(STDIN));
    echo "Enter length: ";
    $l = trim(fgets(STDIN));
    echo "Enter height: ";
    $h = trim(fgets(STDIN));

    //Add each new object to the array
    $list_of_obj[$i] = new Box($w, $l, $h);
}

for ($i = 0; $i <= BOXES - 1; $i++) {
    $list_of_obj[$i]->display_dimensions();
    $list_of_obj[$i]->display_volume();
}

?>

```

5. Solution

```

<?php
class Box {
    private $_width;
    private $_length;
    private $_height;

    //Define the constructor
    function __construct($w, $l, $h) {
        //Initialize fields (using the corresponding set methods)
        $this->setWidth($w);
        $this->setLength($l);
        $this->setHeight($h);
    }

    //Define the getter
    function getWidth() {
        return $this->_width;
    }

    //Define the setter
    function setWidth($value) {
        if ($value > 0) {
            $this->_width = $value;
        }
        else {
            throw new Exception("Cannot be negative or zero");
        }
    }
}

```

```

    }

    //Define the getter
    function getLength() {
        return $this->_length;
    }

    //Define the setter
    function setLength($value) {
        if ($value > 0) {
            $this->_length = $value;
        }
        else {
            throw new Exception("Cannot be negative or zero");
        }
    }

    //Define the getter
    function getHeight() {
        return $this->_height;
    }

    //Define the setter
    function setHeight($value) {
        if ($value > 0) {
            $this->_height = $value;
        }
        else {
            throw new Exception("Cannot be negative or zero");
        }
    }

    function display_volume() {
        echo "Volume: ", ($this->getWidth() * $this->getLength() * $this->getHeight()), "\n";
    }

    function display_dimensions() {
        echo $this->getWidth(), " x ", $this->getLength(), " x ", $this->getHeight(), "\n";
    }
}

define("BOXES", 3);

$list_of_obj = []; //Create an array

for ($i = 0; $i <= BOXES - 1; $i++) {
    echo "Enter width: ";
    $w = trim(fgets(STDIN));
    echo "Enter length: ";
    $l = trim(fgets(STDIN));
    echo "Enter height: ";
    $h = trim(fgets(STDIN));

```

```

        //Add each new object to the array
        $list_of_obj[$i] = new Box($w, $l, $h);
    }

    for ($i = 0; $i <= BOXES - 1; $i++) {
        $list_of_obj[$i]->display_dimensions();
        $list_of_obj[$i]->display_volume();
    }
?>

```

6. Solution

```

<?php
class Cube {
    private $_edge;

    //Define the constructor
    function __construct($edge) {
        $this->_edge = $edge;
    }

    function display_volume() {
        echo "Volume: ", $this->_edge ** 3, "\n";
    }

    function display_one_surface() {
        echo "One surface: ", $this->_edge ** 2, "\n";
    }

    function display_total_surface() {
        echo "Total surface: ", 6 * $this->_edge ** 2, "\n";
    }
}

echo "Enter edge length of a cube: ";
$edge = trim(fgets(STDIN));

$cube1 = new Cube($edge);

$cube1->display_volume();
$cube1->display_one_surface();
$cube1->display_total_surface();
?>

```

7. Solution

```

<?php
class Cube {
    private $_edge;

    //Define the constructor
    function __construct($edge) {
        $this->setEdge($edge);
    }
}

```

```

//Define the getter
function getEdge() {
    return $this->_edge;
}

//Define the setter
function setEdge($value) {
    if ($value > 0) {
        $this->_edge = $value;
    }
    else {
        throw new Exception("Cannot be negative or zero");
    }
}

function display_volume() {
    echo "Volume: ", $this->getEdge() ** 3, "\n";
}

function display_one_surface() {
    echo "One surface: ", $this->getEdge() ** 2, "\n";
}

function display_total_surface() {
    echo "Total surface: ", 6 * $this->getEdge() ** 2, "\n";
}
}

echo "Enter edge length of a cube: ";
$edge = trim(fgets(STDIN));

$cubel = new Cube($edge);

$cubel->display_volume();
$cubel->display_one_surface();
$cubel->display_total_surface();
?>

```

8. Solution

```

<?php
class Circle {
    private $_radius = -1;

    //Define the getter
    function getRadius() {
        if ($this->_radius > 0) {
            return $this->_radius;
        }
        else {
            throw new Exception("Radius is not set");
        }
    }
}

```

```
}

//Define the setter
function setRadius($value) {
    if ($value > 0) {
        $this->_radius = $value;
    }
    else {
        throw new Exception("Cannot be negative or zero");
    }
}

function get_diameter() {
    return 2 * $this->getRadius();
}

function get_area() {
    return 3.14 * $this->getRadius() ** 2;
}

function get_perimeter() {
    return 2 * 3.14 * $this->getRadius();
}
}

function display_menu() {
    echo "1. Enter radius\n";
    echo "2. Display radius\n";
    echo "3. Display diameter\n";
    echo "4. Display area\n";
    echo "5. Display perimeter\n";
    echo "6. Exit\n";
}

$circle1 = new Circle();

while (true) {
    display_menu();
    echo "Enter a choice: ";
    $choice = trim(fgets(STDIN));

    if ($choice == 6) {
        echo "Bye\n";
        break;
    }
    elseif ($choice == 1) {
        echo "Enter radius: ";
        $radius = trim(fgets(STDIN));
        $circle1->setRadius($radius);
    }
    elseif ($choice == 2) {
        echo "Radius: ", $circle1->getRadius(), "\n";
    }
}
```

```

elseif ($choice == 3) {
    echo "Diameter: ", $circle1->get_diameter(), "\n";
}
elseif ($choice == 4) {
    echo "Area: ", $circle1->get_area(), "\n";
}
elseif ($choice == 5) {
    echo "Perimeter: ", $circle1->get_perimeter(), "\n";
}
}
?>

```

9. Solution

```

<?php
class Info {
    private $_user_text;

    //Define the getter
    function getUser_text() {
        return $this->_user_text;
    }

    //Define the setter
    function setUser_text($value) {
        if ($value != "") {
            $this->_user_text = $value;
        }
        else {
            throw new Exception("Cannot be set to empty");
        }
    }

    function get_spaces_count() {
        $count = 0;
        for ($i = 0; $i <= strlen($this->getUser_text()) - 1; $i++) {
            $character = $this->getUser_text()[$i];
            if ($character == " ") {
                $count += 1;
            }
        }
        return $count;
    }

    function get_words_count() {
        return $this->get_spaces_count() + 1;
    }

    function get_vowels_count() {
        $count = 0;
        for ($i = 0; $i <= strlen($this->getUser_text()) - 1; $i++) {
            $character = strtolower($this->getUser_text()[$i]);

```

```

        if (strpos("aeiou", $character) !== false) {
            $count += 1;
        }
    }
    return $count;
}

function get_letters_count() {
    return strlen($this->getUser_text()) - $this->get_spaces_count();
}
}

$inf = new Info();

echo "Enter a text: ";
$inf->setUser_text(trim(fgets(STDIN)));

echo "Text: ", $inf->getUser_text(), "\n";
echo "Spaces: ", $inf->get_spaces_count(), "\n";
echo "Words: ", $inf->get_words_count(), "\n";
echo "Vowels: ", $inf->get_vowels_count(), "\n";
echo "Total number of letters: ", $inf->get_letters_count(), "\n";
?>

```

10. Solution

```

<?php
define("alphabet", " abcdefghijklmnopqrstuvwxyz"); //space is a valid character!

class EncryptDecrypt {
    private $_encr_decr_key = -1;

    //Define the getter
    function getEncr_decr_key() {
        if ($this->_encr_decr_key != -1) {
            return $this->_encr_decr_key;
        }
        else {
            throw new Exception("Key is not set");
        }
    }

    //Define the setter
    function setEncr_decr_key($value) {
        if ($value >= 1 && $value <= 26) {
            $this->_encr_decr_key = $value;
        }
        else {
            throw new Exception("Must be between 1 and 26");
        }
    }

    function encrypt($message) {

```

```

$return_value = "";
for ($i = 0; $i <= strlen($message) - 1; $i++) {
    $character = $message[$i];
    $index = strpos($alphabet, $character);
    $new_index = $index + $this->getEncr_decr_key();
    if ($new_index >= 27) {
        $new_index -= 27;
    }
    $new_letter = $alphabet[$new_index];
    $return_value .= $new_letter;
}
return $return_value;
}

function decrypt($enc_message) {
    $return_value = "";
    for ($i = 0; $i <= strlen($enc_message) - 1; $i++) {
        $character = $enc_message[$i];
        $index = strpos($alphabet, $character);
        $new_index = $index - $this->getEncr_decr_key();
        if ($new_index < 0) {
            $new_index += 27;
        }
        $new_letter = $alphabet[$new_index];
        $return_value .= $new_letter;
    }
    return $return_value;
}
}

function display_menu() {
    echo "1. Enter encryption/decryption key\n";
    echo "2. Encrypt a message\n";
    echo "3. Decrypt a message\n";
    echo "4. Exit\n";
}

$ed = new EncryptDecrypt();

display_menu();
echo "Enter a choice: ";
$choice = trim(fgets(STDIN));
while ($choice != 4) {
    if ($choice == 1) {
        echo "Enter encryption/decryption key: ";
        $ed->setEncr_decr_key(trim(fgets(STDIN)));
    }
    elseif ($choice == 2) {
        echo "Enter message to encrypt: ";
        $text = trim(fgets(STDIN));
        echo "Encrypted message: ", $ed->encrypt($text), "\n";
    }
}

```

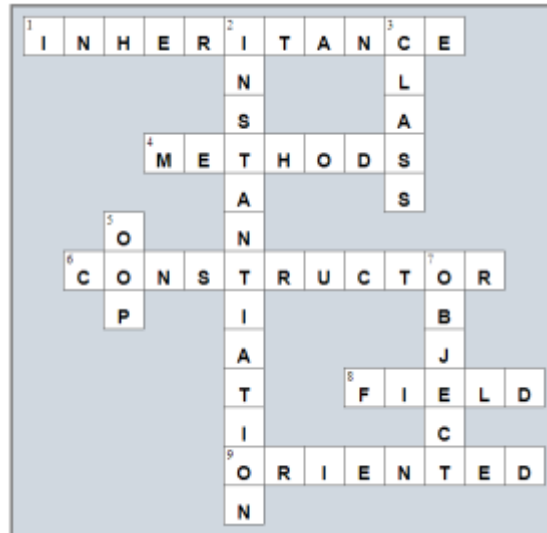


```
elseif ($choice == 3) {  
    echo "Enter message to decrypt: ";  
    $text = trim(fgets(STDIN));  
    echo "Decrypted message: ", $ed->decrypt($text), "\n";  
}  
  
display_menu();  
echo "Enter a choice: ";  
$choice = trim(fgets(STDIN));  
}  
?>
```

Review in “Object Oriented Programming”

Review Crossword Puzzle

1.



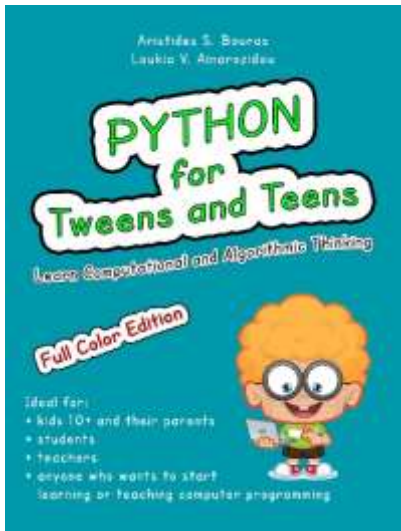
Some Final Words from the Author

I hope you really enjoyed reading this book. I made every possible effort to make it comprehensible even by people that probably have no previous experience in programming.

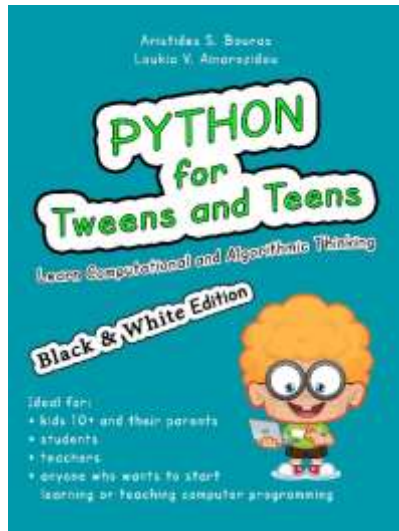
So if you liked this book, please visit the web store where you bought it and show me your gratitude by writing a good review and giving me as many stars as possible. By doing this, you will encourage me to continue writing and of course you'll help other readers to reach me.

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

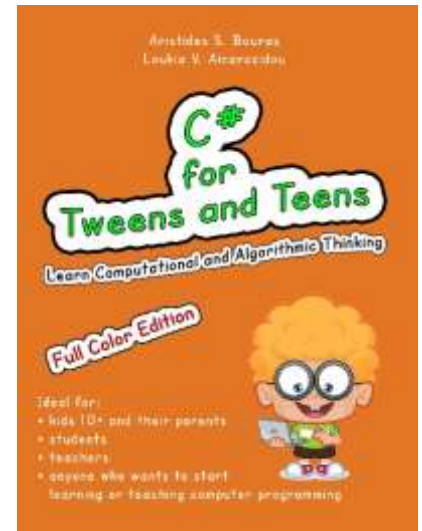
Some of my Books



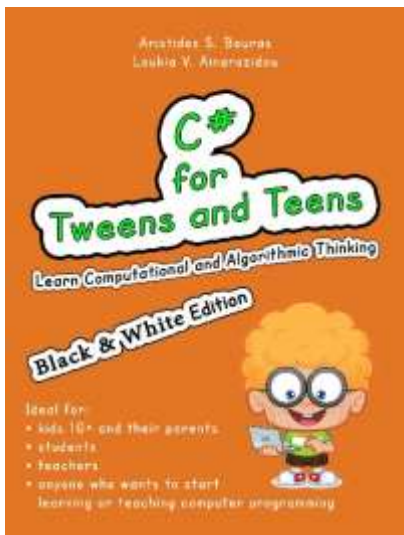
ISBN-10: 1543127940
ISBN-13: 978-1543127942



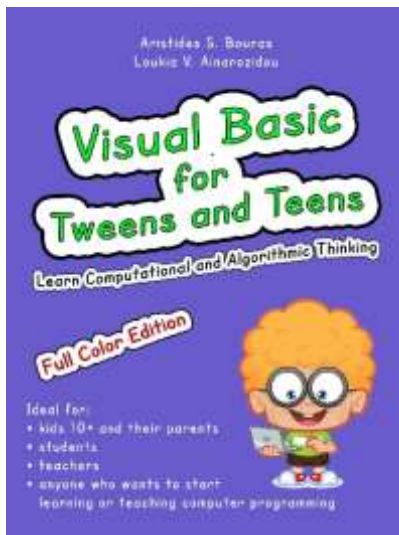
ISBN-10: 1546611215
ISBN-13: 978-1546611219



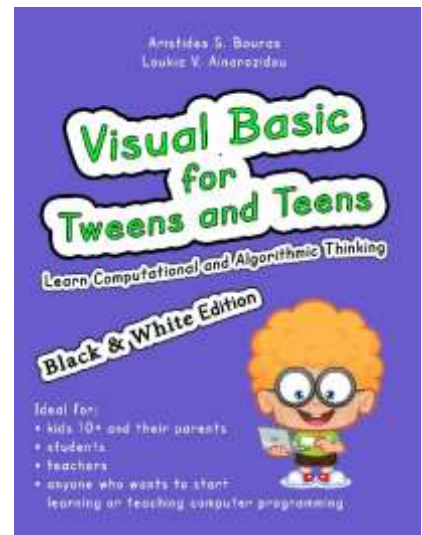
ISBN-10: 1973727684
ISBN-13: 978-1973727682



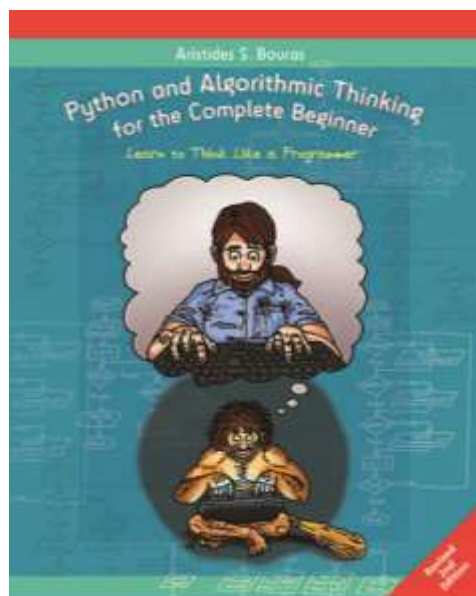
ISBN-10: 1973727765
ISBN-13: 978-1973727767



ISBN-10: 1982083670
ISBN-13: 978-1982083670

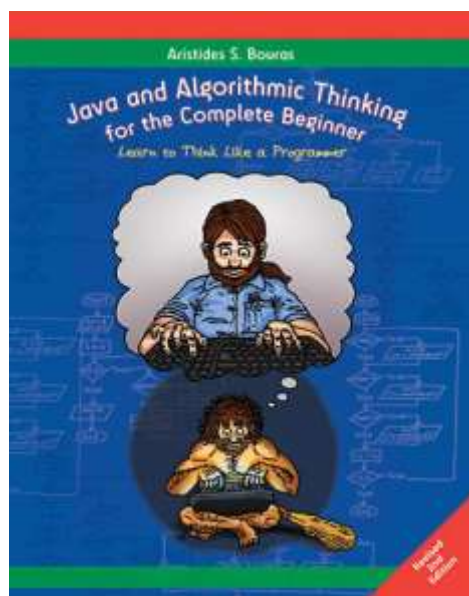


ISBN-10: 1982083697
ISBN-13: 978-1982083694



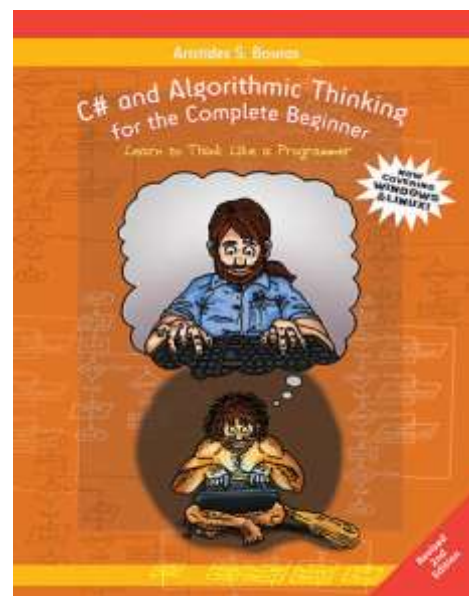
ISBN-10: 1099184878

ISBN-13: 978-1099184871



ISBN-10: 1698811233

ISBN-13: 978-1698811239



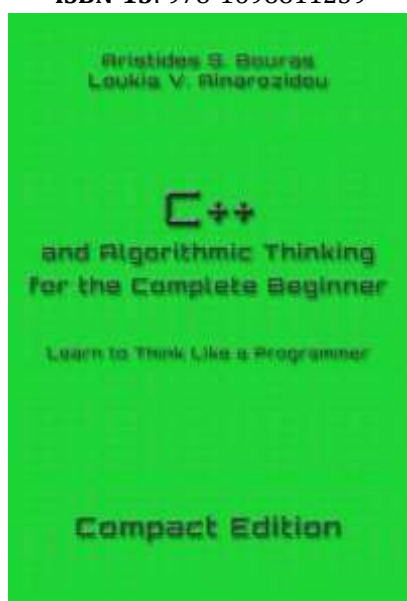
ISBN-10: 1671594363

ISBN-13: 978-1671594364



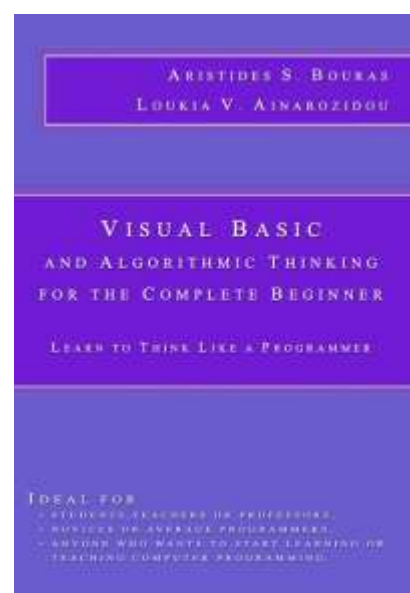
ISBN-10: 1508577552

ISBN-13: 978-1508577553



ISBN-10: 1726791068

ISBN-13: 978-1726791069



ISBN-10: 1511798963

ISBN-13: 978-1511798969