Solutions Companion

Python
AND ALGORITHMIC THINKING
FOR THE COMPLETE BEGINNER

3rd Revised Edition

Aristides S. Bouras

https://www.bouraspage.com

RCode: 221008

Warning and Disclaimer

This book is designed to provide the answers to all of the review questions, as well as the solutions to all review exercises of the book "PYTHON AND ALGORITHMIC THINKING FOR THE COMPLETE BEGINNER – 3rd Revised Edition". Every effort has been taken to make this book compatible with all releases of Python 3.x, and it is almost certain to be compatible with any future releases of Python.

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

Table of Contents

How to Report Errata

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

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

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

If you Like the Book

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

Chapter 11

1.7 Review Questions: True/False

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

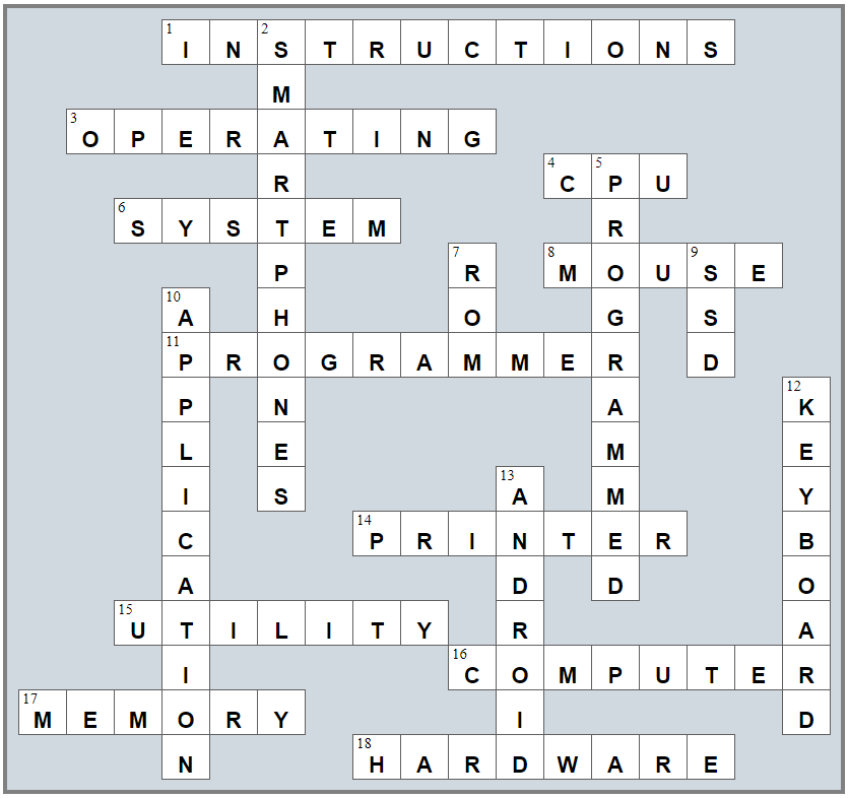
1.8 Review Questions: Multiple Choice

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

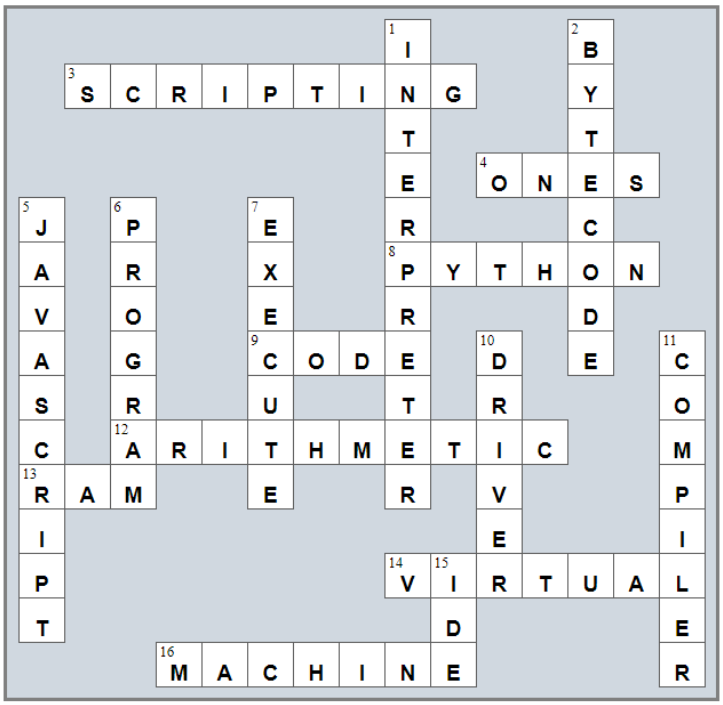
Review in “Introductory Knowledge”

Review Crossword Puzzles

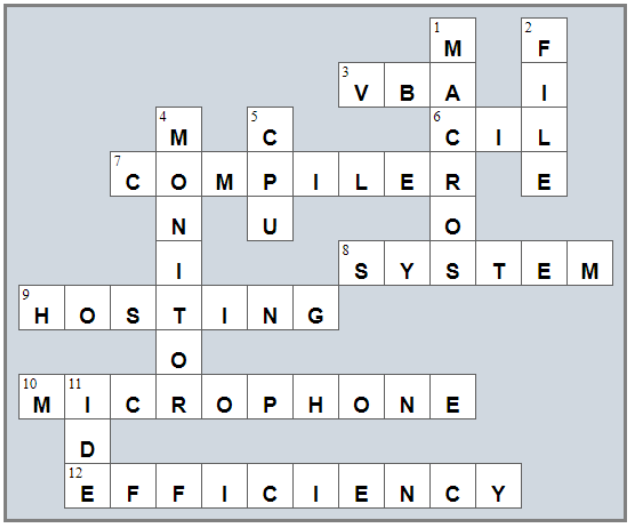
1.



2.



3.


1 M
2 F
3 V B A I
4 M
5 C
6 C I L
7 C O M P I L E R E
N U O
I
8 S Y S T E M
9 H O S T I N G
O
10 M 11 I C R O P H O N E
D
12 E F F I C I E N C Y

Chapter 4

4.17 Review Questions: True/False

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

4.18 Review Questions: Multiple Choice

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

Chapter 5

5.8 Review Questions: True/False

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

5.9 Review Questions: Multiple Choice

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

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

Chapter 6

6.4 Review Questions: True/False

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

6.5 Review Questions: Multiple Choice

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

Chapter 7

7.6 Review Questions: True/False

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

7.7 Review Questions: Multiple Choice

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

7.8 Review Exercises

1. ii, iv, v, ix, x
2. i. String, ii. Boolean, iii. String, iv. String, v. Float, vi. Integer
3. i. d, ii. f, iii. c, iv. e
4. i. 26, ii. 28
5. i. 5, ii. 6
6. i. 1, ii. 0, iii. 1, iv. 1, v. 0, vi. 1
7. i. 2 * 3, ii. 4.0
8. i. 2, ii. 0, iii. 1, iv. 0, v. 0, vi. 0
9. i. 2.0, ii. 5.0
10. My name is George Malkovich
11. i. (−3), ii. 1
12. California California California

Chapter 8

8.2 Review Questions: True/False

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

8.3 Review Exercises

1. Solution

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

2. Solution

For the input value of 3

Step	Statement	a	b	c	d
1	`a = int(input())`	**3**	?	?	?
2	`a = (a + 1) * (a + 1) + 6 / 3 * 2 + 20`	**40.0**	?	?	?
3	`b = a % 13`	40.0	**1.0**	?	?
4	`c = b % 7`	40.0	1.0	**1.0**	?
5	`d = a * b * c`	40.0	1.0	1.0	**40.0**
6	`print(a, ",", b, ",", c, ",", d)`	It displays: 40.0, 1.0, 1.0, 40.0			

For the input value of 4

Step	Statement	a	b	c	d
1	`a = int(input())`	**4**	?	?	?
2	`a = (a + 1) * (a + 1) + 6 / 3 * 2 + 20`	**49.0**	?	?	?
3	`b = a % 13`	49.0	**10.0**	?	?
4	`c = b % 7`	49.0	10.0	**3.0**	?
5	`d = a * b * c`	49.0	10.0	3.0	**1470.0**
6	`print(a, ",", b, ",", c, ",", d)`	It displays: 49.0, 10.0, 3.0, 1470.0			

For the input value of 1

Step	Statement	a	b	c	d
1	`a = int(input())`	**1**	?	?	?
2	`a = (a + 1) * (a + 1) + 6 / 3 * 2 + 20`	**28.0**	?	?	?
3	`b = a % 13`	28.0	**2.0**	?	?
4	`c = b % 7`	28.0	2.0	**2.0**	?
5	`d = a * b * c`	28.0	2.0	2.0	**112.0**
6	`print(a, ",", b, ",", c, ",", d)`	It displays: 28.0, 2.0, 2.0, 112.0			

3. *Solution*

For the input values of 8, 4

Step	Statement	a	b	c	d	e
1	`a = int(input())`	**8**	?	?	?	?
2	`b = int(input())`	8	**4**	?	?	?
3	`c = a + b`	8	4	**12**	?	?
4	`d = 1 + a / b * c + 2`	8	4	12	**27.0**	?
5	`e = c + d`	8	4	12	27.0	**39.0**
6	`c += d + e`	8	4	**78.0**	27.0	39.0
7	`e -= 1`	8	4	78.0	27.0	**38.0**
8	`d -= c + d % c`	8	4	78.0	**–78.0**	38.0
9	`print(c, ",", d, ",", e)`	It displays: 78.0, –78.0, 38.0				

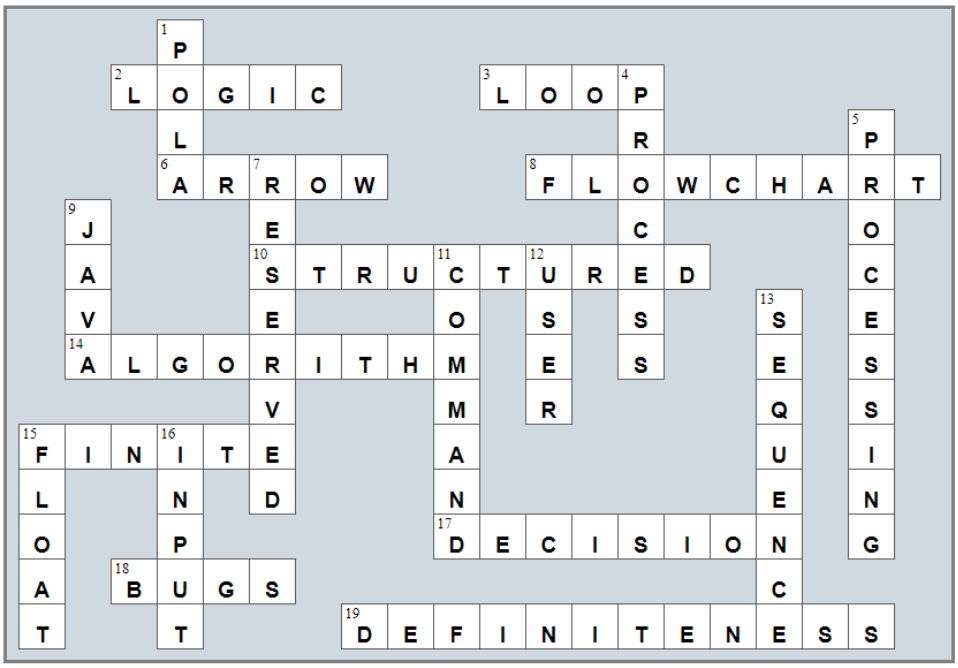
For the input values of 4, 4

Step	Statement	a	b	c	d	e
1	`a = int(input())`	**4**	?	?	?	?
2	`b = int(input())`	4	**4**	?	?	?
3	`c = a + b`	4	4	**8**	?	?
4	`d = 1 + a / b * c + 2`	4	4	8	**11.0**	?
5	`e = c + d`	4	4	8	11.0	**19.0**
6	`c += d + e`	4	4	**38.0**	11.0	19.0
7	`e -= 1`	4	4	38.0	11.0	**18.0**
8	`d -= c + d % c`	4	4	38.0	**–38.0**	18.0
9	`print(c, ",", d, ",", e)`	It displays: 38.0, –38.0, 18.0				

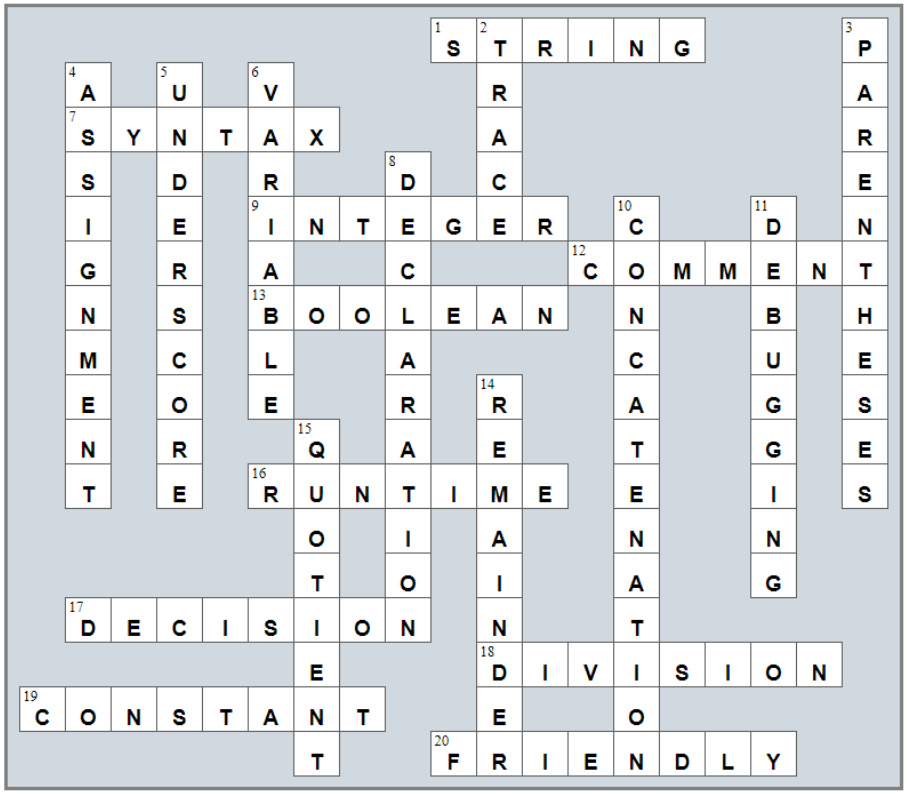
Review in “Getting Started with Python”

Review Crossword Puzzles

1.



2.



Chapter 10

10.2 Review Exercises

1. Solution

```
milesDriven = float(input("Enter miles driven: "))
gallons = float(input("Enter gallons of gas used: "))

mpg = milesDriven / gallons

print("Your car's MPG is:", mpg)
```

2. Solution

```
b = float(input("Enter base: "))
h = float(input("Enter height: "))

area = 0.5 * b * h

print(area)
```

3. Solution

```
angle1 = float(input("Enter 1st angle: "))
angle2 = float(input("Enter 2nd angle: "))

angle3 = 180 - angle1 - angle2

print(angle3)
```

4. Solution

```
g1 = int(input("Enter 1st grade: "))
g2 = int(input("Enter 2nd grade: "))
g3 = int(input("Enter 3rd grade: "))
g4 = int(input("Enter 4th grade: "))

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

print(average)
```

5. Solution

```
PI = 3.14159

r = float(input("Enter radius: "))

perimeter = 2 * PI * r
```

```
print(perimeter)
```

6. Solution

```
PI = 3.14159

d = float(input("Enter diameter (in meters): "))

volume = 4 / 3 * PI * (d / 2) ** 3

print(volume)
```

7. Solution

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

8. Solution

```
firstName = input("First name: ")
middleName = input("Middle name: ")
lastName = input("Last name: ")
title = input("Title: ")

print(title, firstName, middleName, lastName)
print(firstName, middleName, lastName)
print(lastName + ", " + firstName)
print(lastName + ", " + firstName, middleName)
print(lastName + ", " + firstName, middleName + ", " + title)
print(firstName, lastName)
```

9. Solution

```
PI = 3.14159

d = float(input("Enter diameter: "))

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

print(radius, perimeter, area, volume)
```

10. Solution

```
charge = float(input("Enter charge for a meal: "))

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

total = charge + tip + tax
```

```
print(total)
```

11. Solution

```
s = float(input("Enter the distance traveled (in meters): "))
minutes = int(input("Enter the minutes traveled: "))
seconds = int(input("Enter the seconds traveled: "))

totalSeconds = minutes * 60 + seconds

a = 2 * s / totalSeconds ** 2

print(a)
```

12. Solution

```
f = float(input("Enter temperature in Fahrenheit: "))

c = 5 / 9 * (f - 32)

print(c)
```

13. Solution

```
w = int(input("Enter weight in pounds: "))
h = int(input("Enter height in inches: "))

bmi = w * 703 / h ** 2

print(bmi)
```

14. Solution

```
sTotal = float(input("Enter subtotal: "))
gRate = int(input("Enter gratuity rate (0 - 100): "))

tip = sTotal * gRate / 100

total = sTotal + tip

print("Tip is $", tip, " and total is $", total , sep = "")
```

15. Solution

```
VAT = 0.20

btaxPrice1 = float(input("Enter before-tax price 1: "))
btaxPrice2 = float(input("Enter before-tax price 2: "))
btaxPrice3 = float(input("Enter before-tax price 3: "))
```

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

avg = (ataxPrice1 + ataxPrice2 + ataxPrice3) / 3

print(avg)
```

16. Solution

```
VAT = 0.20

ataxPrice = float(input("Enter after-tax price: "))

btaxPrice = ataxPrice / (1 + VAT)

print(btaxPrice)
```

17. Solution

```
iPrice = float(input("Enter price: "))
discount = int(input("Enter discount (0 - 100): "))

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

print(fPrice, saved)
```

18. Solution

```
VAT = 0.20

iKWh = int(input("Enter kWh at the beginning of the month: "))
fKWh = int(input("Enter kWh at the end of the month: "))

kWhConsumed = fKWh - iKWh

cost = kWhConsumed * 0.06
cost += cost * VAT

print(kWhConsumed, cost)
```

19. Solution

```
soldYachts = int(input("Enter number of yachts sold: "))

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

```
print(totalEarnings - totalCost)
```

20. Solution

```
month = int(input("Enter current month: "))
day = int(input("Enter current day: "))

daysPassed = (month - 1) * 30 + day

print(daysPassed)
```

21. Solution

```
month = int(input("Enter current month: "))
day = int(input("Enter current day: "))

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

print(daysLeft)
```

Chapter 11

11.3 Review Questions: True/False

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

11.4 Review Questions: Multiple Choice

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

11.5 Review Exercises

1. Solution

For the input value of 9

Step	Statement	a	b	c
1	`a = float(input())`	**9.0**	?	?
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	`print(a, ",", b, ",", c)`	It displays: 33.4, 1, 1		

For the input value of 4

Step	Statement	a	b	c
1	`a = float(input())`	**4.0**	?	?
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	`print(a, ",", b, ",", c)`	It displays: 30.4, 2, 2		

2. Solution

For the input value of −2

Step	Statement	a	b	c
1	`a = int(input())`	**−2**	?	?
2	`b = abs(a) % 4 + a ** 4`	−2	**18**	**?**
3	`c = b % 5`	−2	18	**3**
4	`print(b, ",", c)`	It displays: 18, 3		

For the input value of –3

Step	Statement	a	b	c
1	`a = int(input())`	**–3**	?	?
2	`b = abs(a) % 4 + a ** 4`	–3	**84**	**?**
3	`c = b % 5`	–3	84	**4**
4	`print(b, ",", c)`	It displays: 84, 4		

3. Solution

```
from math import pi

radians = float(input("Enter angle in radians: "))

degrees = radians * 180 / pi

print(degrees)
```

4. Solution

```
from math import sqrt

a = float(input("Enter right angle side A of a right-angled triangle: "))
b = float(input("Enter right angle side B of a right-angled triangle: "))

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

print(hypotenuse)
```

5. Solution

```
from math import tan, pi

th = float(input("Enter angle θ (in degrees) of a right-angled triangle: "))
adjacent = float(input("Enter length of adjacent side: "))

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

print(opposite)
```

Chapter 12

12.2 Review Exercises

1. Solution

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

2. Solution

i. `y = (x + 3) ** (5 * w) / (7 * (x - 4))`
ii. `y = (3 * x ** 2 - x ** 3 / 4) ** (1 / 5)`
iii. `y = 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

```
from math import sqrt

x = float(input("Enter value for x: "))

y = sqrt(x ** 2 + 1) * (x ** 3 + x ** 2)

print(y)
```

4. Solution

```
x = float(input("Enter value for x: "))

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

print(y)
```

5. Solution

```
from math import tan

x = float(input("Enter value for x: "))
w = float(input("Enter value for w: "))

y = x ** (x + 1) / (tan(2 * w / 3 + 5) + tan(x / 2 + 1)) ** 3

print(y)
```

6. Solution

```
x = float(input("Enter value for x: "))
w = float(input("Enter value for w: "))

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

print(y)
```

7. Solution

```
from math import sin, sqrt

x = float(input("Enter value for x: "))
w = float(input("Enter value for w: "))

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

print(y)
```

8. Solution

```
from math import sqrt

a = float(input("Enter length A: "))
b = float(input("Enter length B: "))
c = float(input("Enter length C: "))

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

print(area)
```

Chapter 13

13.2 Review Exercises

1. Solution

```
n = int(input("Enter an integer: "))

lastDigit = n % 10
result = lastDigit * 8

print(result)
```

2. Solution

```
number = int(input("Enter a five-digit integer: "))

digit5 = number % 10
r = number // 10

digit4 = r % 10
r = r // 10

digit3 = r % 10
r = r // 10

digit2 = r % 10
digit1 = r // 10

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

print(number, "+", reversedNumber, "=", number + reversedNumber)
```

However, using the `divmod()` function it can become:

```
number = int(input("Enter a five-digit integer: "))

r, digit5 = divmod(number, 10)
r, digit4 = divmod(r, 10)
r, digit3 = divmod(r, 10)
digit1, digit2 = divmod(r, 10)

reversedNumber = digit5 * 10000 + digit4 * 1000 + digit3 * 100 + digit2 * 10 + digit1
print(number, "+", reversedNumber, "=", number + reversedNumber)
```

3. Solution

```
n = int(input("Enter an integer: "))

result = n % 2
```

```
print(result)
```

4. Solution

```
n = int(input("Enter an integer: "))

result = 1 - n % 2

print(result)
```

5. Solution

```
number = int(input("Enter an elapsed time in seconds: "))

weeks = number // 604800       # 60 * 60 * 24 * 7 = 604800
r = number % 604800

days = r // 86400              # 60 * 60 * 24 = 86400
r = r % 86400

hours = r // 3600
r = r % 3600

minutes = r // 60
seconds = r % 60

print(weeks, "week(s)", days, "day(s)", hours, "hour(s)", minutes, "minute(s) and", seconds, "second(s)")
```

However, using the `divmod()` function it can become:

```
number = int(input("Enter a period of time in seconds: "))

weeks, r = divmod(number, 604800)      # 60 * 60 * 24 * 7 = 604800
days, r = divmod(r, 86400)             # 60 * 60 * 24 = 86400
hours, r = divmod(r, 3600)
minutes, seconds = divmod(r, 60)

print(weeks, "week(s)", days, "day(s)", hours, "hour(s)", minutes, "minute(s) and", seconds, "second(s)")
```

6. Solution

```
amount = int(input("Enter amount of money to withdraw: "))

usd20, r = divmod(amount, 20)
usd10, r = divmod(r, 10)
usd5, usd1 = divmod(r, 5)

print(usd20, "note(s) of $20", usd10, "note(s) of $10", usd5, "note(s) of $5 and", usd1, "note(s) of $1")
```

7. Solution

```
steps = int(input("Enter number of steps: "))
```

```
distance = steps * 25

miles, r = divmod(distance, 63360)
yards, r = divmod(r, 36)
feet, inches = divmod(r, 12)

print(miles, "mile(s)", yards, "yard(s)", feet, "foot/feet and", inches, "inch(es)")
```

Chapter 14

14.4 Review Questions: True/False

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

14.5 Review Questions: Multiple Choice

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

14.6 Review Exercises

1. Solution

```
from random import randrange

alphabet = "abcdefghijklmnopqrstuvwxyz"

randomWord = alphabet[randrange(26)].upper() + alphabet[randrange(26)] + alphabet[randrange(26)] +   \
             alphabet[randrange(26)] + alphabet[randrange(26)]

print(randomWord)
```

2. Solution

```
from random import randrange

name = input("Enter name: ")

x = name.lower().replace(" ", "")

secretPassword = x[randrange(len(x))] + x[randrange(len(x))] + x[randrange(len(x))] +    \
                 str(randrange(1000, 10000))

print(secretPassword)
```

3. Solution

First approach

```
number = int(input("Enter a three-digit integer: "))

#Convert the number to string
```

```
numberString = str(number)

#Reverse the string
reversedString = numberString[2] + numberString[1] + numberString[0]

#Convert the reversed string to integer
reversedNumber = int(reversedString)

print(reversedNumber)
```

Second approach

```
number = int(input("Enter an integer: "))

#Convert the number to string
numberString = str(number)

#Reverse the string
reversedString = numberString[::-1]

#Convert the reversed string to integer
reversedNumber = int(reversedString)

print(reversedNumber)
```

The advantage of this approach is that the user is allowed to enter any integer, no matter how small or large!

Third approach

```
number = int(input("Enter an integer: "))

reversedNumber = int(str(number)[::-1])

print(reversedNumber)
```

The advantage of this approach is that the user is allowed to enter any integer, no matter how small or large!

4. Solution

```
firstName = input("First name: ")
middleName = input("Middle name: ")
lastName = input("Last name: ")

firstName = firstName[0].upper() + firstName[1:].lower()
middleName = middleName[0].upper() + middleName[1:].lower()
lastName = lastName[0].upper() + lastName[1:].lower()

print(firstName, middleName, lastName)
print(firstName, middleName[0] + ".", lastName)
print(lastName, firstName[0] + ".")
```

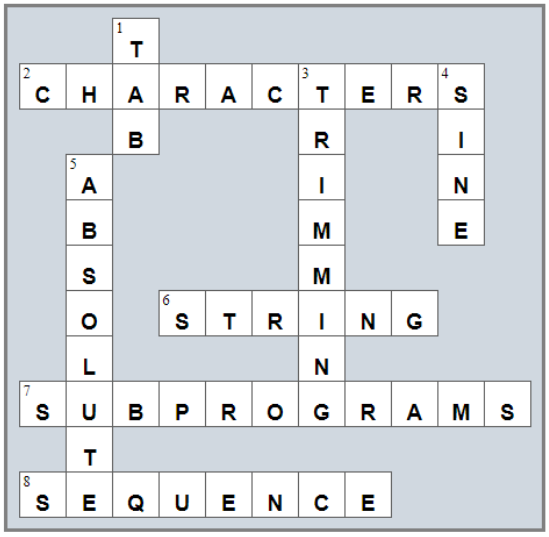
5. Solution

```
word = input("Enter a long word: ")
abbreviation = word[0] + str(len(word) - 2) + word[-1]
print(abbreviation)
```

Review in “Sequence Control Structures”

Review Crossword Puzzle

1.



Chapter 15

15.10 Review Questions: True/False

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

15.11 Review Questions: Multiple Choice

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

15.12 Review Exercises

1. Solution

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

2. Solution

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

3. Solution

BE1 (Boolean Expression 1)	BE2 (Boolean Expression 2)	BE1 or BE2	BE1 and BE2	not(BE2)
False	**False**	False	False	True
False	**True**	True	False	False
True	**False**	True	False	True
True	**True**	True	True	False

4. Solution

a	b	c	a > 3 or c > b and c > 1	a > 3 and c > b or c > 1
4	**−6**	**2**	True	True
−3	**2**	**−4**	False	False
2	**5**	**5**	False	True

5. Solution

Expression	Value
`(x + y) ** 3`	8
`(x + y) / (x ** 2 - 14)`	1.0
`x - 1 == y + 5`	True
`x > 2 and y == 1`	False
`x == 1 or y == -2 and not(flag == False)`	True
`not(x >= 3) and (x % 2 > 1)`	False

6. Solution

i. False
ii. True

7. Solution

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

8. Solution

i. `x != 4 or y == 3`
ii. `x + 4 > 0`
iii. `not(x <= 5) and y != 4`
iv. `x == False`
v. `not(x < 4 and z <= 4)`
vi. `x == 2 or x < -5`

9. Solution

i. `not(x < 4 or y == 10)`
ii. `not(x - 2 < 9)`

iii. `not(not(x < 2) and y == 4)`

iv. `not(x == False and y != 3)`

v. First approach: `not(not(x < 2 or y < 2))`
Second approach: `x < 2 or y < 2`

vi. `not(x == -2 or x > 2)`

Chapter 16

16.2 Review Questions: True/False

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

16.3 Review Questions: Multiple Choice

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

16.4 Review Exercises

1. Solution

The corrections/additions are in red

```
x = float(input())

y = - 5
if x * y / 2 > 20:
  y *= 2
  x += 4 * x ** 2

print(x, y)
```

2. Solution

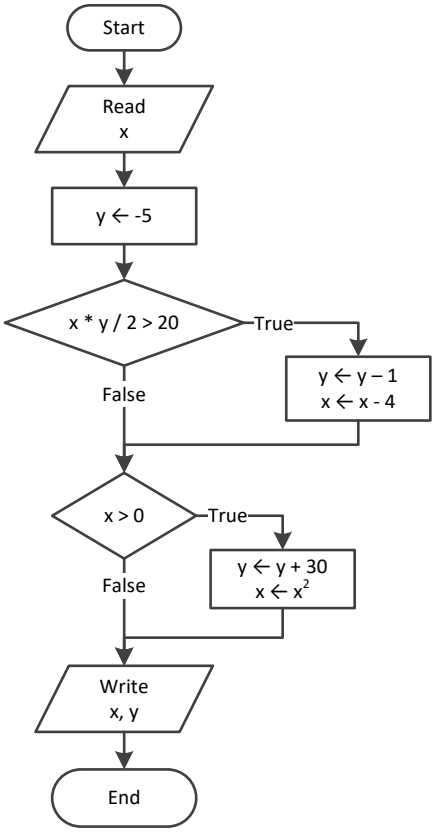
For the input value of 10

Step	Statement	x	y
1	x = float(input())	**10.0**	?
2	y = -5	10.0	**–5**
3	if x * y / 2 > 20:	False	
4	if x > 0:	True	
5	y += 30	10.0	**25**
6	x = x ** 2	**100.0**	25
7	print(x, ",", y)	It displays: 100.0, 25	

For the input value of –10

Step	Statement	x	y
1	x = float(input())	**–10.0**	?
2	y = -5	–10.0	**–5**

3	`if x * y / 2 > 20:`	True	
4	`y -= 1`	−10.0	**−6**
5	`x -= 4`	**−14.0**	−6
6	`if x > 0:`	False	
7	`print(x, ",", y)`	It displays: −14.0, −6	



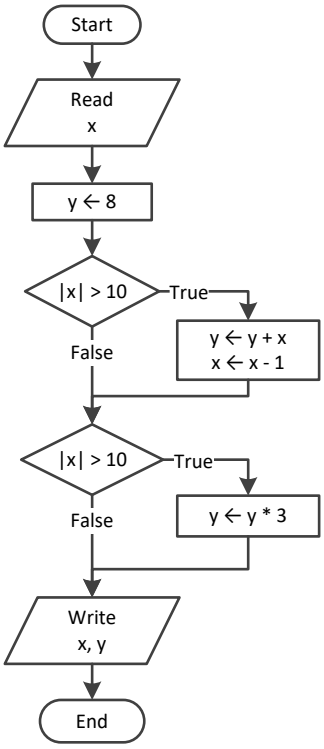
3. *Solution*

For the input value of −11

Step	Statement	x	y
1	`x = int(input())`	**−11**	?
2	`y = 8`	−11	**8**
3	`if abs(x) > 10:`	True	
4	`y += x`	−11	**−3**
5	`x -= 1`	**−12**	−3
6	`if abs(x) > 10:`	True	
7	`y *= 3`	−12	**−9**
8	`print(x, ",", y)`	It displays: −12, −9	

For the input value of 11

Step	Statement	x	y
1	`x = int(input())`	**11**	?
2	`y = 8`	11	**8**
3	`if abs(x) > 10:`	True	
4	`y += x`	11	**19**
5	`x -= 1`	**10**	19
6	`if abs(x) > 10:`	False	
7	`print(x, ",", y)`	It displays: 10, 19	



4. *Solution*

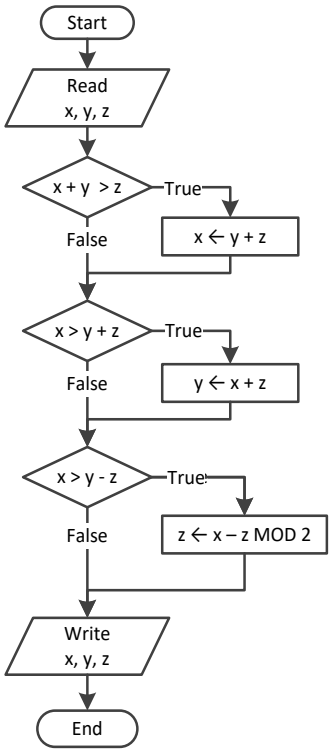
For input values of 1, 2 and 3

Step	Statement	x	y	z
1	`x = int(input())`	**1**	?	?
2	`y = int(input())`	1	**2**	?
3	`z = int(input())`	1	**2**	**3**
4	`if x + y > z:`	False		
5	`if x > y + z:`	False		
6	`if x > y - z:`	True		
7	`z = x - z % 2`	1	2	**0**

8	`print(x, ",", y, ",", z)`	It displays: 1, 2, 0

For input values of 4, 2 and 1

Step	Statement	x	y	z
1	`x = int(input())`	**4**	?	?
2	`y = int(input())`	4	**2**	?
3	`z = int(input())`	4	**2**	**1**
4	`if x + y > z:`		True	
5	`x = y + z`	**3**	2	1
6	`if x > y + z:`		False	
7	`if x > y - z:`		True	
8	`z = x - z % 2`	3	2	**2**
9	`print(x, ",", y, ",", z)`	It displays: 3, 2, 2		



5. Solution

```
x = float(input("Enter a number: "))
if x > 0:
    print("Positive")
```

6. Solution

```
x = float(input("Enter a number: "))
y = float(input("Enter a second number"))
```

```
if x > 0 and y > 0:
    print("Both Positives")
```

7. Solution

```
x = int(input("Enter your age: "))

if x > 14:
    print("You can drive a car in Kansas (USA)")
```

8. Solution

```
s = input("Enter a string: ")

if s == s.upper():
    print("Uppercase")
```

9. Solution

```
s = input("Enter a string: ")

if len(s) > 20:
    print("Many characters")
```

10. Solution

```
n1 = float(input("Enter 1st number: "))
n2 = float(input("Enter 2nd number: "))
n3 = float(input("Enter 3rd number: "))
n4 = float(input("Enter 4th number: "))

if n1 < 0 or n2 < 0 or n3 < 0 or n4 < 0:
    print("Among the provided numbers, there is a negative one!")
```

11. Solution

```
a = float(input("Enter 1st number: "))
b = float(input("Enter 2nd number: "))

if a > b:
    c = a           # Or you can do the following:
    a = b           # a, b = b, a
    b = c           #

print(a, b)
```

12. Solution

```
t1 = float(input("Enter 1st temperature: "))
t2 = float(input("Enter 2nd temperature: "))
```

```
t3 = float(input("Enter 3rd temperature: "))

average = (t1 + t2 + t3) / 3

if average > 60:
    print("Heat Wave")
```

Chapter 17

17.2 Review Questions: True/False

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

17.3 Review Questions: Multiple Choice

1. b
2. c
3. c

17.4 Review Exercises

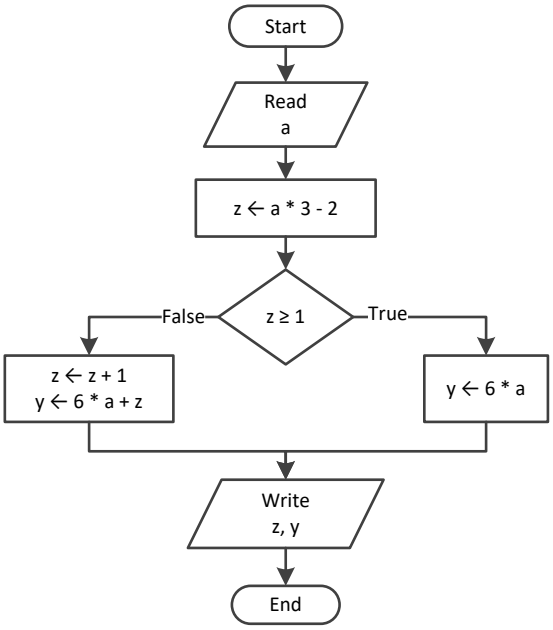
1. Solution

For input value of 3

Step	Statement	a	y	z
1	`a = float(input())`	**3.0**	?	?
2	`z = a * 3 - 2`	3.0	?	**7.0**
3	`if z >= 1:`		True	
4	`y = 6 * a`	3.0	**18.0**	7.0
5	`print(z, ",", y)`	It displays: 7.0, 18.0		

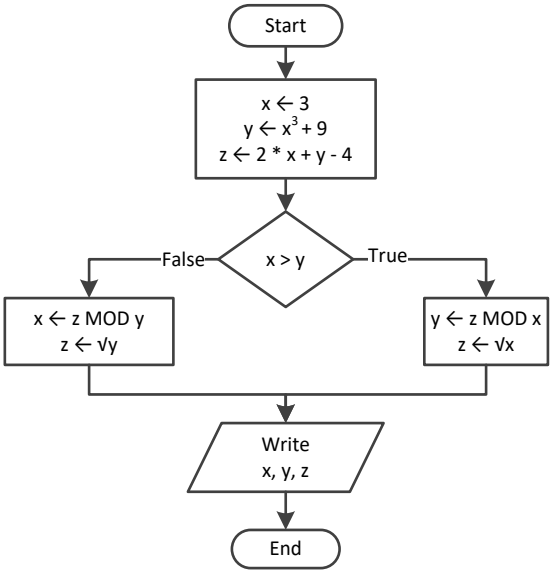
For input value of 0.5

Step	Statement	a	y	z
1	`a = float(input())`	**0.5**	?	?
2	`z = a * 3 - 2`	0.5	?	**–0.5**
3	`if z >= 1:`		False	
4	`z += 1`	0.5	?	**0.5**
5	`y = 6 * a + z`	0.5	**3.5**	0.5
6	`print(z, ",", y)`	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 = sqrt(y)`	2	36	**6**
7	`print(x, ",", y, ",", z)`	It displays: 2, 36, 6		



3. Solution

```
x = float(input())
w = x * 3 - 15
z = (w + 7) * (x + 4) - 10
if w > x and z > x:
    x += 1
    y = x / 2 + 4
else:
    y = x / 4 + 2
print(y)
```

For input value of 10

Step	Statement	x	y	w	z
1	x = float(input())	**10.0**	?	?	?
2	w = x * 3 - 15	10.0	?	**15.0**	?
3	z = (w + 7) * (x + 4) - 10	10.0	?	15.0	**298.0**
4	if w > x and z > x:	True			
5	x += 1	**11.0**	?	15.0	298.0
6	y = x / 2 + 4	11.0	**9.5**	15.0	298.0
7	print(y)	It displays: 9.5			

For input value of 2

Step	Statement	x	y	w	z
1	x = float(input())	**2.0**	?	?	?
2	w = x * 3 - 15	2.0	?	**–9.0**	?
3	z = (w + 7) * (x + 4) - 10	2.0	?	–9.0	**–22.0**
4	if w > x and z > x:	False			
5	y = x / 4 + 2	2.0	**2.5**	–9.0	–22.0
6	print(y)	It displays: 2.5			

4. Solution

```
num = float(input("Enter a number: "))
if num > 100:
    print("Provided number is greater than 100")
else:
    print("Provided number is less than or equal to 100")
```

5. Solution

```
num = float(input("Enter a number: "))
if num >= 0 and num <= 100:
    print("Provided number is between 0 and 100")
else:
```

```
    print("Provided number is not between 0 and 100")
```

6. Solution

```
name1 = input("Enter team name 1: ")
name2 = input("Enter team name 2: ")
goals1 = int(input("Enter goals " + name1 + " scored: "))
goals2 = int(input("Enter goals " + name2 + " scored: "))

if goals1 > goals2:
    print("Winner:", name1)
else:
    print("Winner:", name2)
```

7. Solution

```
x = int(input())
if x % 6 == 0:
    print(x, "is a multiple of 6")
else:
    print(x, "is not a multiple of 6")
```

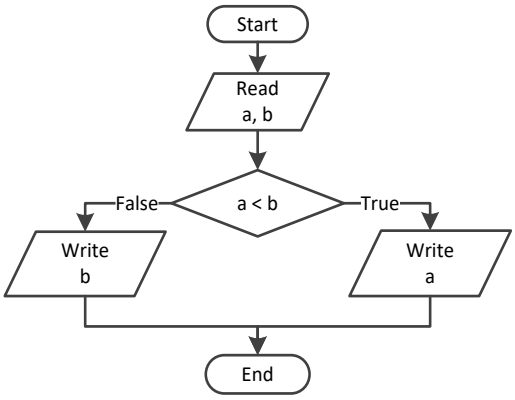
8. Solution

```
x = int(input())
if x % 6 == 0 or x % 7 == 0:
    print(x, "is a multiple of 6 or a multiple of 7")
else:
    print(x, "is neither a multiple of 6 nor a multiple of 7")
```

9. Solution

```
x = int(input())

y = x % 4

if y == 0:
    print(x, "is a multiple of 4")
else:
    print(x, "is not a multiple of 4")

print("The structure is:", x, "=", x // 4, "x 4 +", y)
```

10. Solution




```
a = float(input())
b = float(input())

if a < b:
    print(a)
else:
    print(b)
```

11. Solution

```
a = float(input())
b = float(input())
c = float(input())

if a < b + c and b < a + c and c < a + b:
    print("Provided numbers can be lengths of the three sides of a triangle")
else:
    print("Provided numbers cannot be lengths of the three sides of a triangle")
```

12. Solution

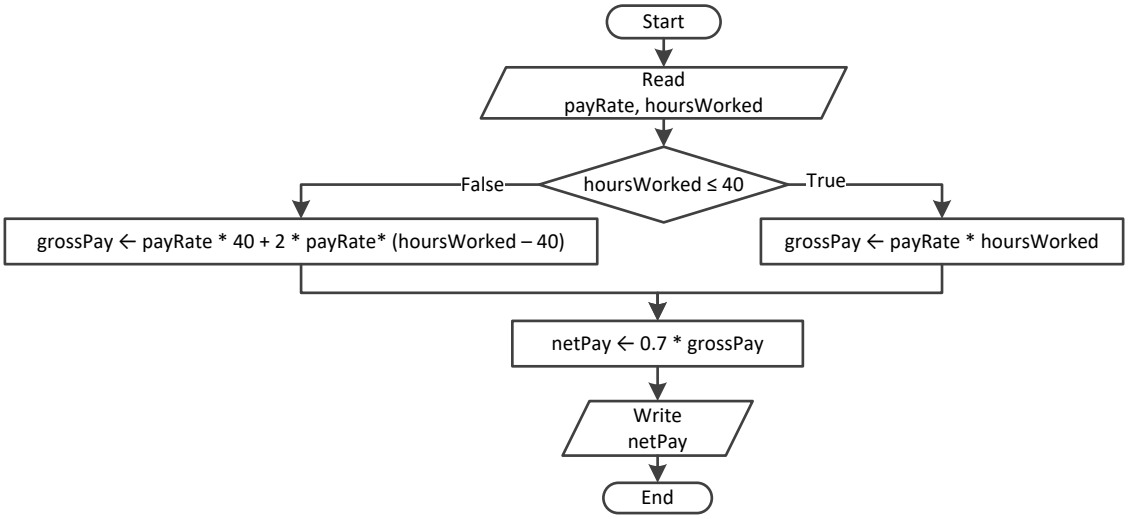
```
a = float(input())
b = float(input())
c = float(input())

if a ** 2 == b ** 2 + c ** 2 or b ** 2 == a ** 2 + c ** 2 or c ** 2 == a ** 2 + b ** 2:
    print("Provided numbers can be lengths of the three sides of a right triangle")
else:
    print("Provided numbers cannot be lengths of the three sides of a right triangle")
```

13. Solution

```
a = float(input("Enter 1st jump in meters: "))
b = float(input("Enter 2nd jump in meters: "))
c = float(input("Enter 3rd jump in meters: "))
```

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

if average >= 8:
    print("Qualified")
else:
    print("Disqualified")
```

14. Solution




```
payRate = float(input())
hoursWorked = int(input())

if hoursWorked <= 40:
    grossPay = payRate * hoursWorked
else:
    grossPay = payRate * 40 + 2 * payRate * (hoursWorked - 40)

netPay = 0.7 * grossPay
print(netPay)
```

15. Solution

```
miles = int(input("Enter miles traveled: "))

r = miles % 12000

if r > 6000:
    milesLeft = 12000 - r
    print("Your car needs a major service in", milesLeft, "miles")
else:
    milesLeft = 6000 - r
    print("Your car needs a minor service in", milesLeft, "miles")
```

16. Solution

```
t = float(input("Enter the time the two cars traveled: "))
a1 = float(input("Enter the accelaration for car A: "))
a2 = float(input("Enter the accelaration for car B: "))

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

print("Distance between them:", abs(s1 - s2), "meters")

if s1 > s2:
    print("Car A is first")
else:
    print("Car B is first")
```

Chapter 18

18.2 Review Questions: True/False

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

18.3 Review Exercises

1. Solution

For input value of 5

Step	Statement	q	b
1	`q = int(input())`	**5**	?
2	`if 0 < q <= 50:`	True	
3	`b = 1`	5	**1**
4	`print(b)`	It displays: 1	

For input value of 150

Step	Statement	q	b
1	`q = int(input())`	**150**	?
2	`if 0 < q <= 50:`	False	
3	`elif 50 < q <= 100:`	False	
4	`elif 100 < q <= 200:`	True	
5	`b = 3`	150	**3**
6	`print(b)`	It displays: 3	

For input value of 250

Step	Statement	q	b
1	`q = int(input())`	**250**	?
2	`if 0 < q <= 50:`	False	
3	`elif 50 < q <= 100:`	False	
4	`elif 100 < q <= 200:`	False	
5	`b = 4`	250	**4**
6	`print(b)`	It displays: 4	

For input value of −1

Step	Statement	q	b
1	`q = int(input())`	**−1**	?
2	`if 0 < q <= 50:`	False	

3	`elif 50 < q <= 100:`	False	
4	`elif 100 < q <= 200:`	False	
5	`b = 4`	–1	**4**
6	`print(b)`	It displays: 4	

2. Solution

For input value of 5

Step	Statement	amount	discount	payment
1	`amount = float(input())`	**5.0**	?	?
2	`discount = 0`	5.0	**0**	?
3	`if amount < 20:`	True		
4	`discount = 0`	5.0	**0**	?
5	`payment = amount - amount * discount / 100`	5.0	0	**5.0**
6	`print(discount, ",", payment)`	It displays: 0, 5.0		

For input value of 150

Step	Statement	amount	discount	payment
1	`amount = float(input())`	**150.0**	?	?
2	`discount = 0`	150.0	**0**	?
3	`if amount < 20:`	False		
4	`elif 20 <= amount < 60:`	False		
5	`elif 60 <= amount < 100:`	False		
6	`elif amount >= 100:`	True		
7	`discount = 15`	150.0	**15**	?
8	`payment = amount - amount * discount / 100`	150.0	15	**127.5**
9	`print(discount, ",", payment)`	It displays: 15, 127.5		

For input value of –1

Step	Statement	amount	discount	payment
1	`amount = float(input())`	**–1.0**	?	?
2	`discount = 0`	–1.0	**0**	?
3	`if amount < 20:`	True		
4	`discount = 0`	–1.0	**0**	?
5	`payment = amount - amount * discount / 100`	–1.0	0	**–1.0**
6	`print(discount, ",", payment)`	It displays: 0, –1.0		

3. Solution

For input value of 1

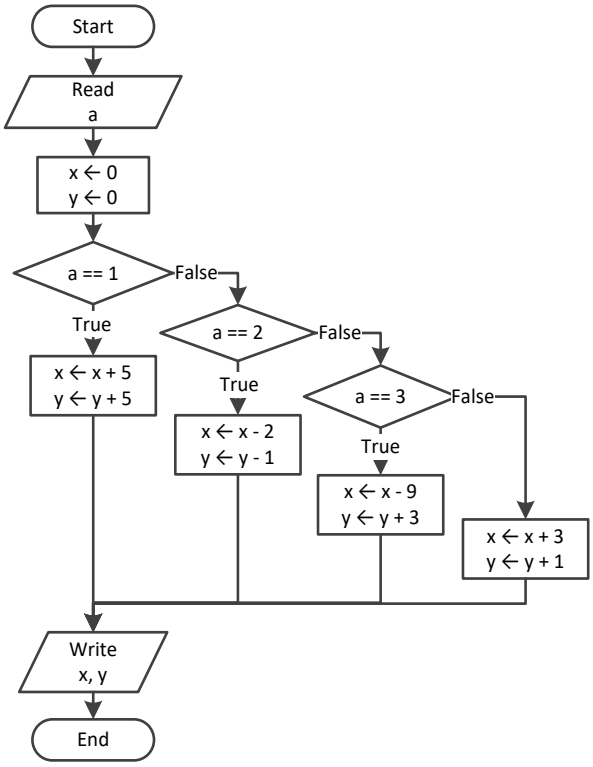
Step	Statement	a	x	y
1	`a = int(input())`	**1**	?	?
2	`x = 0`	1	**0**	?
3	`y = 0`	1	0	**0**
4	`if a == 1:`	True		
5	`x = x + 5`	1	**5**	0
6	`y = y + 5`	1	5	**5**
7	`print(x, ",", y)`	It displays: 5, 5		

For input value of 3

Step	Statement	a	x	y
1	`a = int(input())`	**3**	?	?
2	`x = 0`	3	**0**	?
3	`y = 0`	3	0	**0**
4	`if a == 1:`	False		
5	`elif a == 2:`	False		
6	`elif a == 3:`	True		
7	`x = x - 9`	3	**–9**	0
8	`y = y + 3`	3	–9	**3**
9	`print(x, ",", y)`	It displays: –9, 3		

For input value of 250

Step	Statement	a	x	y
1	`a = int(input())`	**250**	?	?
2	`x = 0`	250	**0**	?
3	`y = 0`	250	0	**0**
4	`if a == 1:`	False		
5	`elif a == 2:`	False		
6	`elif a == 3:`	False		
7	`x = x + 3`	250	**3**	0
8	`y += 1`	250	3	**1**
9	`print(x, ",", y)`	It displays: 3, 1		



4. Solution

For input values of 10, 2, 5

Step	Statement	a	x	y
1	`a = int(input())`	**10**	?	?
2	`x = int(input())`	10	**2**	?
3	`y = float(input())`	10	2	**5.0**
4	`if a == 10:`		True	
5	`x = x % 2`	10	**0**	5.0
6	`y = y ** 2`	10	0	**25.0**
7	`print(x, ",", y)`	It displays: 0, 25.0		

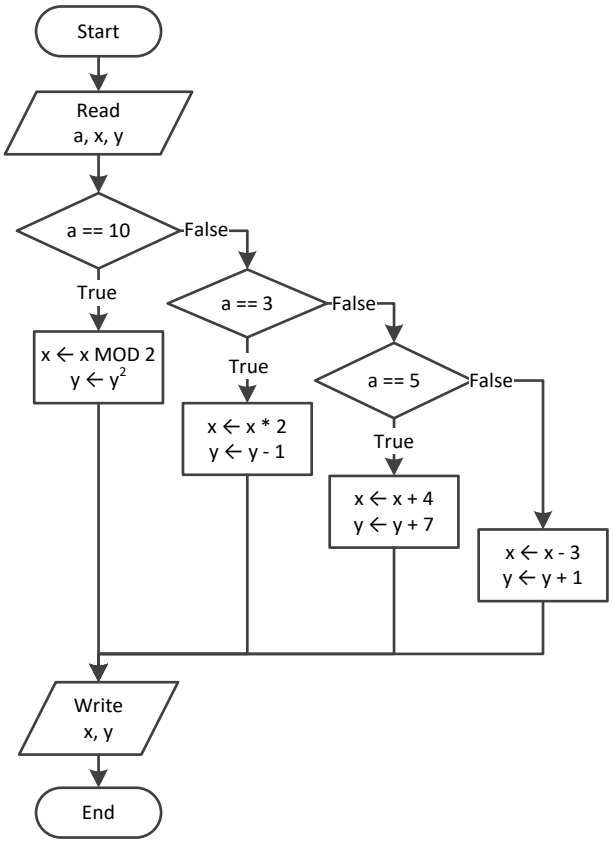
For input values of 5, 2, 3

Step	Statement	a	x	y
1	`a = int(input())`	**5**	?	?
2	`x = int(input())`	5	**2**	?
3	`y = float(input())`	5	2	**3.0**
4	`if a == 10:`		False	
5	`elif a == 3:`		False	
6	`elif a == 5:`		True	

7	`x = x + 4`	5	**6**	3.0
8	`y += 7`	5	6	**10.0**
9	`print(x, ",", y)`	It displays: 6, 10.0		

For input values of 4, 6, 2

Step	Statement	a	x	y
1	`a = int(input())`	**4**	?	?
2	`x = int(input())`	4	**6**	?
3	`y = float(input())`	4	6	**2.0**
4	`if a == 10:`	False		
5	`elif a == 3:`	False		
6	`elif a == 5:`	False		
7	`x -= 3`	4	**3**	2.0
8	`y += 1`	4	3	**3.0**
9	`print(x, ",", y)`	It displays: 3, 3.0		



5. Solution

```
a = float(input())
if a < 1:
    y = 5 + a
```

```
    print(y)
elif a < 5:
    y = 23 / a
    print(y)
elif a < 10:
    y = 5 * a
    print(y)
else:
    print("Error!")
```

6. Solution

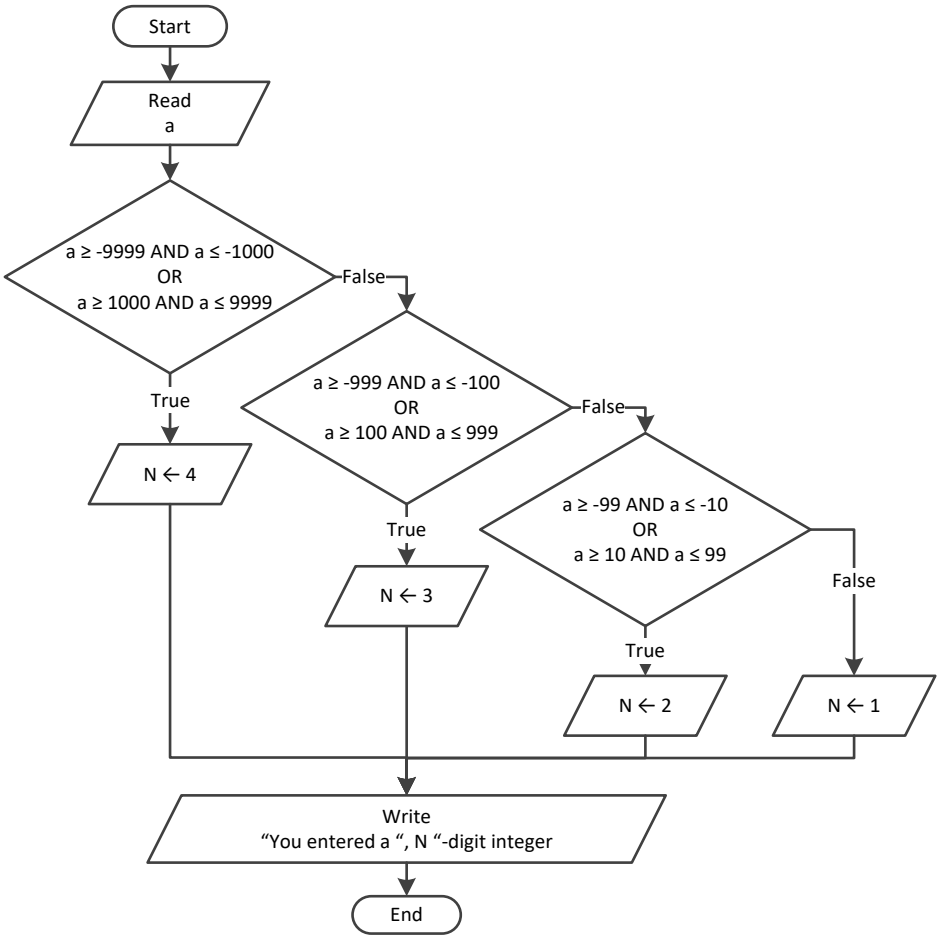
```
n1 = int(input("Enter an integer: "))
n2 = int(input("Enter a second integer: "))

if n1 % 2 == 0 and n2 % 2 == 0:
    print("Both numbers are evens")
elif n1 % 2 != 0 and n2 % 2 != 0:
    print("Both numbers are odds")
else:
    print("Nothing special!")
```

7. Solution

```
name1 = input("Enter team name 1: ")
name2 = input("Enter team name 2: ")
goals1 = int(input("Enter goals " + name1 + " scored: "))
goals2 = int(input("Enter goals " + name2 + " scored: "))

if goals1 > goals2:
    print("Winner:", name1)
elif goals2 > goals1:
    print("Winner:", name2)
else:
    print("It's a tie!")
```

8. Solution

First approach




```
a = int(input())

if -9999 <= a <= -1000 or 1000 <= a <= 9999:
    n = 4
elif -999 <= a <= -100 or 100 <= a <= 999:
    n = 3
elif -99 <= a <= -10 or 10 <= a <= 99:
    n = 2
else:
    n = 1

print("You entered a ", n, "-digit integer", sep = "")
```

Second approach

```
a = abs(int(input()))

if 1000 <= a <= 9999:
    n = 4
elif 100 <= a <= 999:
```

```
    n = 3
elif 10 <= a <= 99:
    n = 2
else:
    n = 1

print("You entered a ", n, "-digit integer", sep = "")
```

Third approach

```
a = int(input())
aString = str(abs(a))
print("You entered a ", len(aString), "-digit integer", sep = "")
```

9. Solution

First approach

```
a = int(input())

if -9999 <= a <= -1000 or 1000 <= a <= 9999:
    print("You entered a 4-digit integer", sep = "")
elif -999 <= a <= -100 or 100 <= a <= 999:
    print("You entered a 3-digit integer", sep = "")
elif -99 <= a <= -10 or 10 <= a <= 99:
    print("You entered a 2-digit integer", sep = "")
elif -9 <= a <= 9:  #Include the value of zero
    print("You entered a 1-digit integer", sep = "")
else:
    print("Error: Invalid value!")
```

Second approach

```
a = abs(int(input()))

if 1000 <= a <= 9999:
    print("You entered a 4-digit integer", sep = "")
elif 100 <= a <= 999:
    print("You entered a 3-digit integer", sep = "")
elif 10 <= a <= 99:
    print("You entered a 2-digit integer", sep = "")
elif 0 <= a <= 9:
    print("You entered a 1-digit integer", sep = "")
else:
    print("Error: Invalid value!")
```

Third approach

```
a = int(input())

if -9999 <= a <= 9999:
    aString = str(abs(a))
    print("You entered a ", len(aString), "-digit integer", sep = "")
else:
    print("Error: Invalid value!")
```

10. Solution

```
print("1. Convert USD to Euro (EUR)")
print("2. Convert USD to British Pound Sterling (GBP)")
print("3. Convert USD to Japanese Yen (JPY)")
print("4. Convert USD to Canadian Dollar (CAD)")
ch = int(input("Enter a choice: "))

usd = float(input("Enter an amount in US dollars: "))

if ch == 1:
    eur = usd * 0.94
    print("$", usd, " = ", eur, " EUR", sep = "")
elif ch == 2:
    gbp = usd * 0.81
    print("$", usd, " = ", gbp, " GBP", sep = "")
elif ch == 3:
    jpy = usd * 149.11
    print("$", usd, " = ", jpy, " JPY", sep = "")
else:
    cad = usd * 1.36
    print("$", usd, " = ", cad, " CAD", sep = "")
```

11. Solution

```
m = int(input("Enter the number of a month between 1 and 12: "))

if m <= 2 or m == 12:
    print("Winter")
elif m <= 5:
    print("Spring")
elif m <= 8:
    print("Summer")
else:
    print("Fall (Autumn)")
```

12. Solution

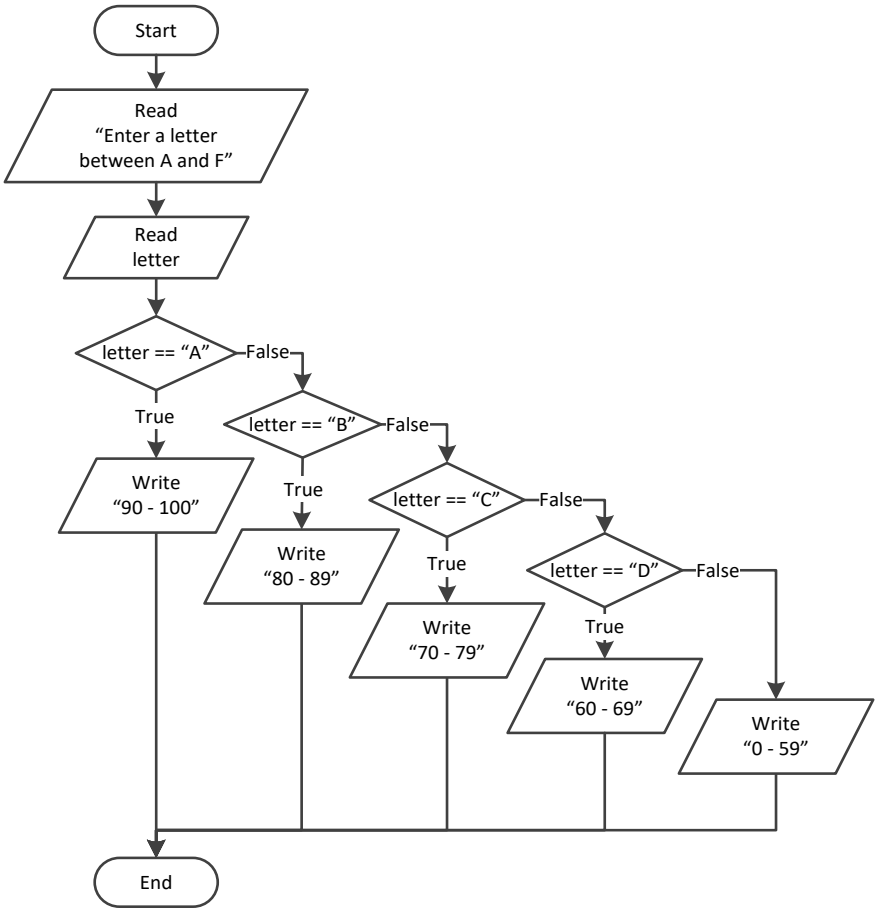
```
m = int(input("Enter the number of a month between 1 and 12: "))

if m < 1 or m > 12:
    print("Error: Invalid value!")
elif m <= 2 or m == 12:
    print("Winter")
elif m <= 5:
    print("Spring")
elif m <= 8:
    print("Summer")
else:
```

```
    print("Fall (Autumn)")
```

13. Solution

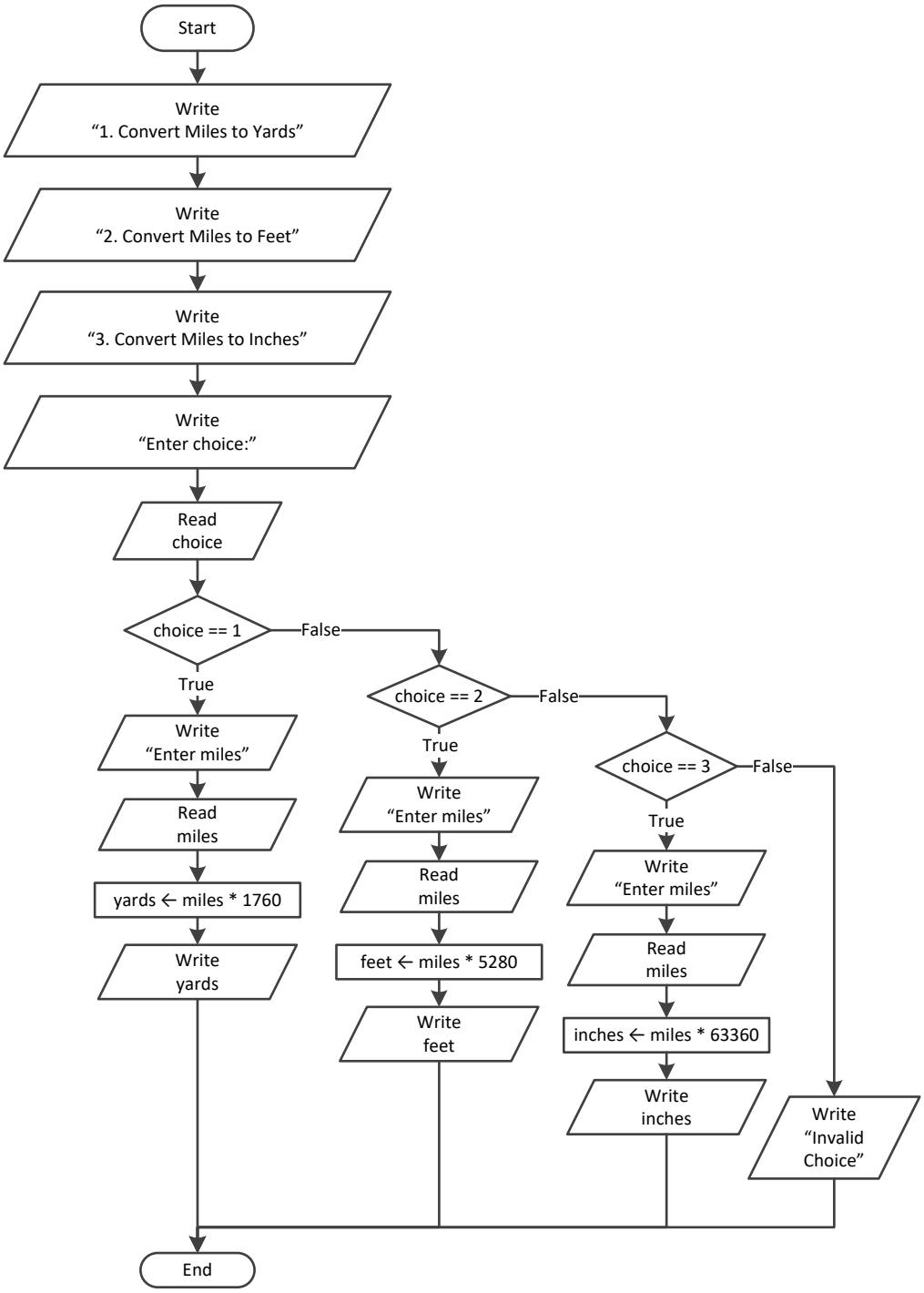
```
name = input("Enter the name of a month: ")

if name == "January":
    print(1)
elif name == "February":
    print(2)
elif name == "March":
    print(3)
elif name == "April":
    print(4)
elif name == "May":
    print(5)
elif name == "June":
    print(6)
elif name == "July":
    print(7)
elif name == "August":
    print(8)
elif name == "September":
    print(9)
elif name == "October":
    print(10)
elif name == "November":
    print(11)
elif name == "December":
    print(12)
else:
    print("Error")
```

14. Solution




```
letter = input("Enter a letter between A and F: ")

if letter == "A":
    print("90 - 100")
elif letter == "B":
    print("80 - 89")
elif letter == "C":
    print("70 - 79")
elif letter == "D":
    print("60 - 69")
else:
    print("0 - 59")
```

15. Solution




```
print("1. Convert Miles to Yards")
print("2. Convert Miles to Feet")
print("3. Convert Miles to Inches")

choice = int(input("Enter a choice: "))
```

```
if choice == 1:
    miles = float(input("Enter miles: "))
    yards = miles * 1760
    print(miles, "miles =", yards, "yards")
elif choice == 2:
    miles = float(input("Enter miles: "))
    feet = miles * 5280
    print(miles, "miles =", feet, "feet")
elif choice == 3:
    miles = float(input("Enter miles: "))
    inches = miles * 63360
    print(miles, "miles =", inches, "inches")
else:
    print("Invalid choice!")
```

16. Solution

```
roman = input("Enter a Roman numeral between I and X: ")

if roman == "I":
    print(1)
elif roman == "II":
    print(2)
elif roman == "III":
    print(3)
elif roman == "IV":
    print(4)
elif roman == "V":
    print(5)
elif roman == "VI":
    print(6)
elif roman == "VII":
    print(7)
elif roman == "VIII":
    print(8)
elif roman == "IX":
    print(9)
elif roman == "X":
    print(10)
else:
    print("Error")
```

17. Solution

```
bottles = int(input("Enter the total number of wine bottles purchased in a month: "))

if bottles == 1:
    print("You are awarded 3 points")
elif bottles == 2:
    print("You are awarded 10 points")
```

```
elif bottles == 3:
    print("You are awarded 20 points")
else:
    print("You are awarded 45 points")
```

18. Solution

```
from random import randrange

name = input("Enter your name: ")

i = randrange(3)

if i == 0:
    print("Hello", name + "!")
elif i == 1:
    print("Hi", name + "!")
elif i == 2:
    print("What's up", name + "!")
```

19. Solution

```
num = input()

if num == "zero":
    print(0)
elif num == "one":
    print(1)
elif num == "two":
    print(2)
elif num == "three":
    print(3)
elif num == "four":
    print(4)
elif num == "five":
    print(5)
elif num == "six":
    print(6)
elif num == "seven":
    print(7)
elif num == "eight":
    print(8)
elif num == "nine":
    print(9)
else:
    print("I don't know this number!")
```

20. Solution

```
b = int(input("Enter Beaufort number: "))
```

```
if b == 0:
    print("Calm")
elif b == 1:
    print("Light Air")
elif b == 2:
    print("Light breeze")
elif b == 3:
    print("Gentle breeze")
elif b == 4:
    print("Moderate breeze")
elif b == 5:
    print("Fresh breeze")
elif b == 6:
    print("Strong breeze")
elif b == 7:
    print("Moderate gale")
elif b == 8:
    print("Gale")
elif b == 9:
    print("Strong gale")
elif b == 10:
    print("Storm")
elif b == 11:
    print("Violent storm")
elif b == 12:
    print("Hurricane force")
else:
    print("Invalid Beaufort number!")
```

21. Solution

```
n = float(input("Enter a number between 0.0 and 9.9: "))

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

number = ""

if x == 1:
    number += "One"
elif x == 2:
    number += "Two"
elif x == 3:
    number += "Three"
elif x == 4:
    number += "Four"
elif x == 5:
    number += "five"
elif x == 6:
```

```
    number += "six"
elif x == 7:
    number += "seven"
elif x == 8:
    number += "eight"
elif x == 9:
    number += "nine"
elif x == 0:
    number += "zero"

number += " point "

if y == 1:
    number += "one"
elif y == 2:
    number += "two"
elif y == 3:
    number += "three"
elif y == 4:
    number += "four"
elif y == 5:
    number += "five"
elif y == 6:
    number += "six"
elif y == 7:
    number += "seven"
elif y == 8:
    number += "eight"
elif y == 9:
    number += "nine"
elif y == 0:
    number += "zero"

print(number)
```

Chapter 19

19.2 Review Questions: True/False

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

19.3 Review Exercises

1. Solution

For input values of 20, 1

Step	Statement	x	y
1	`x = int(input())`	**20**	?
2	`y = int(input())`	20	**1**
3	`if x < 30:`	True	
4	`if y == 1:`	True	
5	`x = x % 3`	**2**	1
6	`y = 5`	2	**5**
7	`print(x, ",", y)`	It displays: 2, 5	

For input values of 20, 3

Step	Statement	x	y
1	`x = int(input())`	**20**	?
2	`y = int(input())`	20	**3**
3	`if x < 30:`	True	
4	`if y == 1:`	False	
5	`elif y == 2:`	False	
6	`elif y == 3:`	True	
7	`x = x + 5`	**25**	3
8	`y += 3`	25	**6**
9	`print(x, ",", y)`	It displays: 25, 6	

For input values of 12, 8

Step	Statement	x	y
1	`x = int(input())`	**12**	?
2	`y = int(input())`	12	**8**
3	`if x < 30:`	True	
4	`if y == 1:`	False	
5	`elif y == 2:`	False	
6	`elif y == 3:`	False	

7	`x -= 2`	**10**	8
8	`y += 1`	10	**9**
9	`print(x, ",", y)`	It displays: 10, 9	

For input values of 50, 0

Step	Statement	x	y
1	`x = int(input())`	**50**	?
2	`y = int(input())`	50	**0**
3	`if x < 30:`	False	
4	`y += 1`	50	**1**
5	`print(x, ",", y)`	It displays: 50, 1	

2. *Solution*

For input values of 60, 25

Step	Statement	x	y
1	`x = int(input())`	**60**	?
2	`y = int(input())`	60	25
3	`if (x + y) / 2 <= 20:`	False	
4	`if y < 15:`	False	
5	`elif y < 23:`	False	
6	`x = 2 * x + 5`	**125**	25
7	`y += 1`	125	**26**
8	`print(x, ",", y)`	It displays: 125, 26	

For input values of 50, 8

Step	Statement	x	y
1	`x = int(input())`	**50**	?
2	`y = int(input())`	50	8
3	`if (x + y) / 2 <= 20:`	False	
4	`if y < 15:`	True	
5	`x = x % 4`	**2**	8
6	`y = 2`	2	**2**
7	`print(x, ",", y)`	It displays: 2, 2	

For input values of 20, 15

Step	Statement	x	y
1	`x = int(input())`	**20**	?
2	`y = int(input())`	20	15
3	`if (x + y) / 2 <= 20:`	True	

4	`if y < 10:`	False	
5	`elif y < 20:`	True	
6	`x = x * 5`	**100**	15
7	`y += 2`	100	**17**
8	`print(x, ",", y)`	It displays: 100, 17	

For input values of 10, 30

Step	Statement	x	y
1	`x = int(input())`	**10**	?
2	`y = int(input())`	10	30
3	`if (x + y) / 2 <= 20:`	True	
4	`if y < 10:`	False	
5	`elif y < 20:`	False	
6	`x = x - 2`	**8**	30
7	`y += 3`	8	**33**
8	`print(x, ",", y)`	It displays: 8, 33	

3. Solution

```
a = int(input())
if a > 1000:
    print("Big Positive")
else:
    if a > 0:
        print("Positive")
    else:
        if a < -1000:
            print("Big Negative")
        else:
            if a < 0:
                print("Negative")
            else:
                print("Zero")
```

4. Solution

First approach

```
age = int(input("Enter your age: "))

if age < 0:
    print("Error: Invalid age!")
else:
    if age < 16:
        print("You cannot drive either a small scooter or a car")
    else:
```

```
        if age < 18:
            print("You can drive a small scooter")
        else:
            print("You can drive a car and a small scooter")
```

Second approach

```
age = int(input("Enter your age: "))

if age < 0:
    print("Error: Invalid age!")
else:
    if age < 16:
        print("You cannot drive either a small scooter or a car")
    elif age < 18:
        print("You can drive a small scooter")
    else:
        print("You can drive a car and a small scooter")
```

Third approach

```
age = int(input("Enter your age: "))

if age < 0:
    print("Error: Invalid age!")
elif age < 16:
    print("You cannot drive either a small scooter or a car")
elif age < 18:
    print("You can drive a small scooter")
else:
    print("You can drive a car and a small scooter")
```

5. Solution

```
soldHoverboards = int(input("Enter number of hoverboards sold: "))
employeesNum = int(input("Enter number of employees: "))

if soldHoverboards < 0 or employeesNum <= 0:
    print("Wrong value(s) entered")
else:
    hoverboardsCost = soldHoverboards * 150
    insuranceCost = employeesNum * 1000
    totalCost = hoverboardsCost + insuranceCost

    totalEarnings = soldHoverboards * 250
    profitLoss = totalEarnings - totalCost

    if profitLoss > 0:
        print("Profit")
    elif profitLoss < 0:
        print("Loss")
    else:
        print("Broke even")
```

6. Solution

First approach: Using nested decision structures

```
from random import randrange

name = input("Enter your name: ")

hour = randrange(1, 25)
print("The hour is", str(hour) + ":00")

if hour >= 5 and hour <= 11:
    print("Good Morning", name + "!");
else:
    if hour >= 12 and hour <= 18:
        print("Good Afternoon", name + "!")
    else:
        if hour >= 19 and hour <= 22:
            print("Good Evening", name + "!")
        else:
            print("Good Night", name + "!")
```

Second approach: Using a multiple-alternative decision structure

```
from random import randrange

name = input("Enter your name: ")

hour = randrange(1, 25)
print("The hour is", str(hour) + ":00")

if hour >= 5 and hour <= 11:
    print("Good Morning", name + "!");
elif hour >= 12 and hour <= 18:
    print("Good Afternoon", name + "!")
elif hour >= 19 and hour <= 22:
    print("Good Evening", name + "!")
else:
    print("Good Night", name + "!")
```

7. Solution

```
print("Enter the three sides of a triangle: ")
a = float(input())
b = float(input())
c = float(input())

if a >= b + c or b >= a + c or c >= a + b:
    print("Provided numbers cannot be lengths of the three sides of a triangle")
else:
    if a == b and b == c:
        print("Equilateral")
    elif a ** 2 == b ** 2 + c ** 2 or b ** 2 == a ** 2 + c ** 2 or c ** 2 == a ** 2 + b ** 2:
```

```
        print("Right (or right-angled)")
    else:
        print("not special")
```

8. Solution

```
pin = int(input("Enter your four-digit PIN : "))
if pin != 1234:
    pin = int(input("Wrong PIN. Enter your four-digit PIN : "))
    if pin != 1234:
        pin = int(input("Wrong PIN. Enter your four-digit PIN : "))

if pin != 1234:
    print("PIN locked!")
else:
    amount = int(input("Enter the amount of money (an integer value) that you want to withdraw: "))
    usd10 = amount // 10
    r = amount % 10
    usd5 = r // 5
    usd1 = r % 5
    print(usd10, "note(s) of $10", usd5, "note(s) of $5", "and", usd1, "note(s) of $1")
```

9. Solution

First approach

```
t = float(input("Enter temperature (in Fahrenheit): "))
w = float(input("Enter wind speed (in miles/hour): "))

if t > 75:
    if w > 12:
        print("The day is hot and windy")
    else:
        print("The day is hot and not windy")
else:
    if w > 12:
        print("The day is cold and windy")
    else:
        print("The day is cold and not windy")
```

Second approach

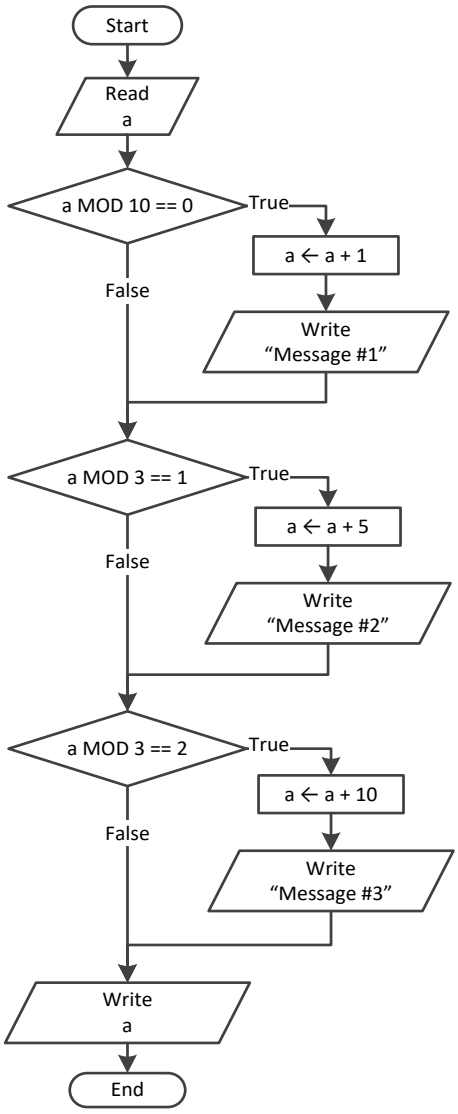
```
t = float(input("Enter temperature (in Fahrenheit): "))
w = float(input("Enter wind speed (in miles/hour): "))

if t > 75:
    message1 = "hot"
else:
    message1 = "cold"

if w > 12:
    message2 = "windy"
```

```
else:
    message2 = "not windy"

print("The day is", message1, "and", message2)
```

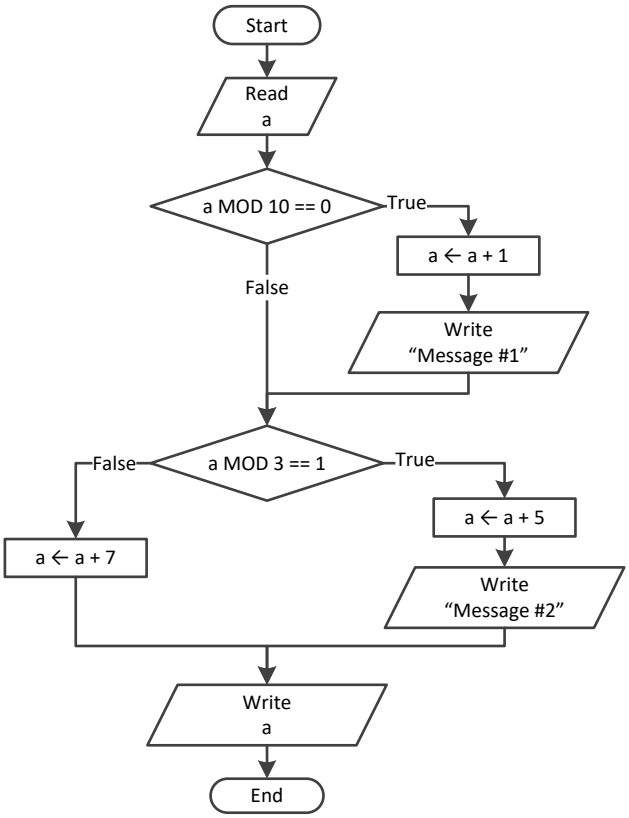
Chapter 20

20.4 Review Exercises

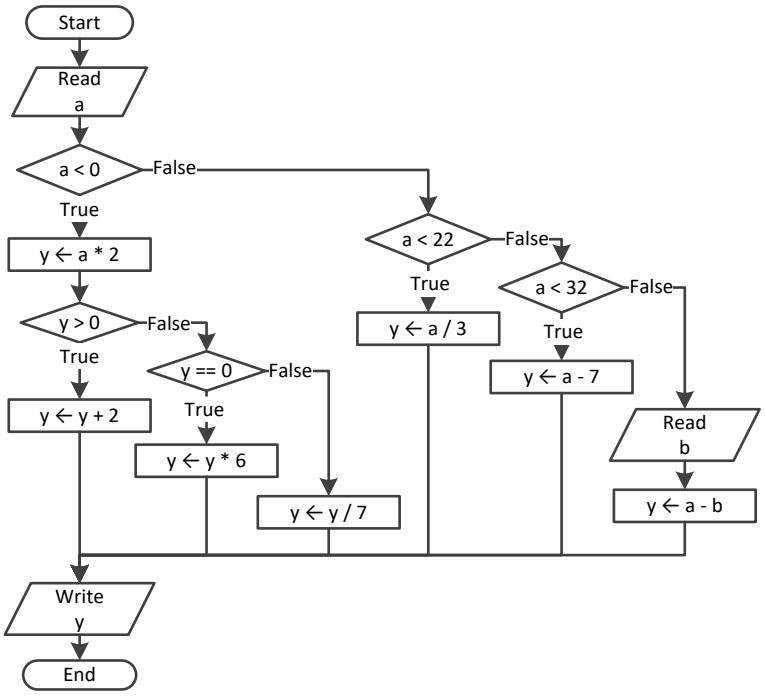
1. Solution



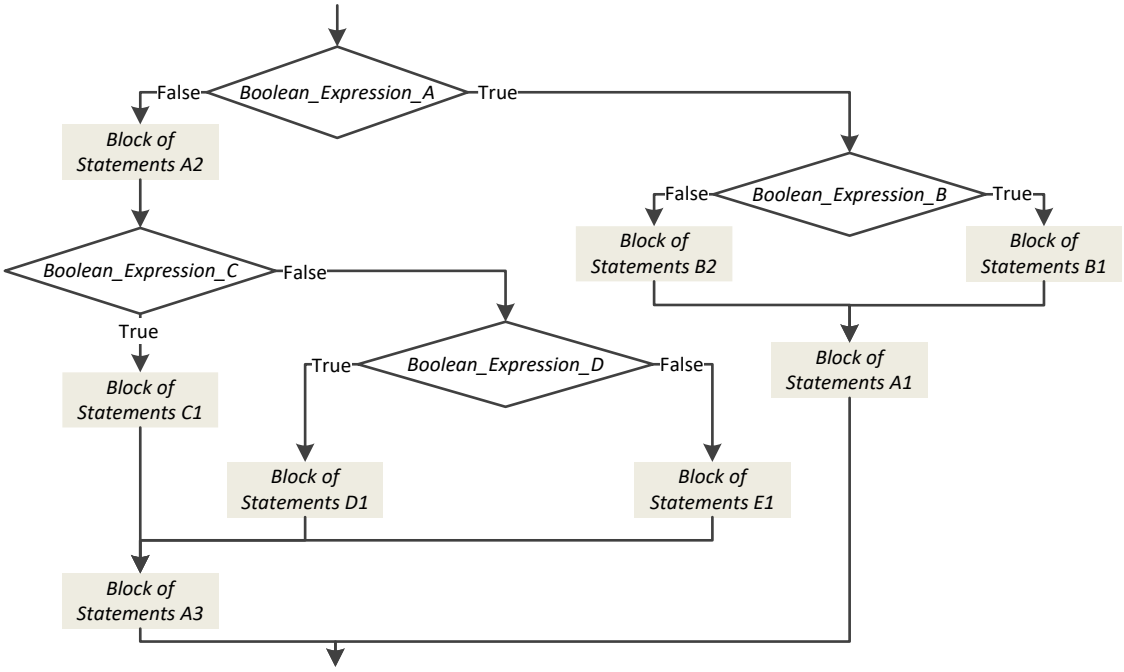
2. Solution



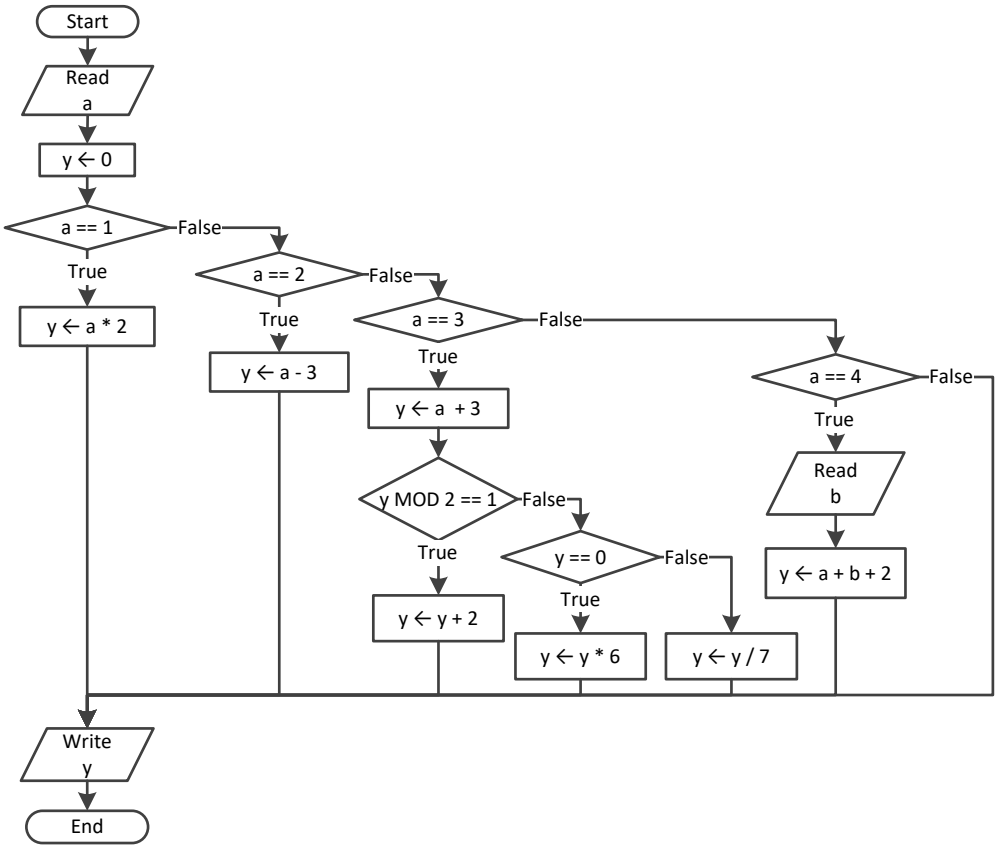
3. Solution



4. Solution


Boolean_Expression_A
False
True
Block of Statements A2
Boolean_Expression_B
False
True
Block of Statements B2
Block of Statements B1
Boolean_Expression_C
False
True
Block of Statements C1
Boolean_Expression_D
True
False
Block of Statements D1
Block of Statements E1
Block of Statements A1
Block of Statements A3

5. Solution



6. Solution

```
x = float(input())
y = float(input())

if x != 100 or y <= 10:
    z = float(input())
    if z <= x + y:
        x -= 3
        y = x + 4
print(x, y)
```

7. Solution

```
x = int(input())

if x == 1:
    print("Good Morning")
    print("How Do you do?")
    print("Is everything okay?")
elif x == 2:
    print("Good Evening")
```

```
    print("How Do you do?")
    print("Is everything okay?")
elif x == 3:
    print("Good Afternoon")
    print("Is everything okay?")
else:
    print("Good Night")
```

8. Solution

```
a = int(input())
b = int(input())

c = a % 2
d = b // 5

if a >= b:
    y = 1
elif d > c and a > 2:
    y = 2
elif d * c > a / b:
    if d * c > 10:
        y = 4
    else:
        y = 3
else:
    y = 5

print(y)
```

9. Solution

```
x = int(input())
if x > 0:
    if x % 10 == 0:
        print("Last digit equal to 0")
    elif x % 10 == 1:
        print("Last digit equal to 1")
    else:
        print("None")
else:
    if x == -1:
        print("Bye")
    else:
        print("Invalid Number")
```

10. Solution

```
a = float(input())
b = float(input())
```

```
y = a * b

if y > 0:
    y -= 1
    y /= 2
else:
    y +=10
    if y > 0:
        y /= 2
    else:
        y *= 2
```

11. Solution

```
a = float(input())
b = float(input())
c = float(input())

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
print(a, b, c)
```

Chapter 21

21.8 Review Questions: True/False

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

21.9 Review Questions: Multiple Choice

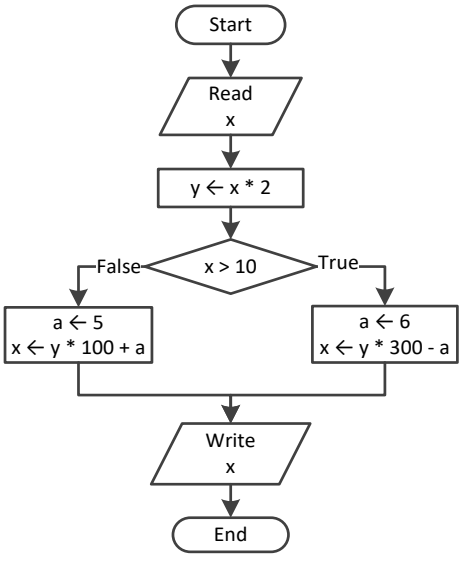
1. a
2. b
3. a

21.10 Review Exercises

1. Solution

```
y = int(input())
x = int(input())

if y > 0:
    a = x * 4 * y + 1
else:
    a = x * 2 * y + 6

print(y)
print(a)
```

2. Solution



3. Solution

```
a = float(input())
```

```
if a >= 10:
    print("Error!")
else:
    if a < 1:
        y = 5 + a
    elif a < 5:
        y = 23 / a
    else:
        y = 5 * a
    print(y)
```

4. Solution

```
day = int(input())
month = int(input())
name = input()

if day == 16 and month == 2 and name == "Loukia":
    print("Happy Birthday!!!")
else:
    print("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

```
a = float(input())
b = float(input())
c = float(input())

if a > 10:
    if c < 2000:
        d = (a + b + c) / 12
        print("The result is:", d)
    else:
        print("Error!")
else:
    print("Error!")
```

6. Solution

```
a = float(input())
b = float(input())
c = float(input())

if a > 10 and b < 2000 and c != 10:
    d = (a + b + c) / 12
    print("The result is:", d)

if a <= 10:
    print("Error!")
```

7. Solution

```
a = int(input())
b = int(input())

y = 3
if a > 0:
    y = y * a
    print("Hello Zeus")

print(y, b)
```

8. Solution

```
a = float(input())
b = float(input())

y = 0
if a > 0:
    y = y + 7
else:
    print("Hello Zeus")
    print(abs(a))
print(y)
```

9. Solution

```
os = input("What is your tablet's OS? ")

if os == "iOS":
    print("Apple")
elif os == "Android":
    print("Google")
elif os == "Windows":
    print("Microsoft")
```

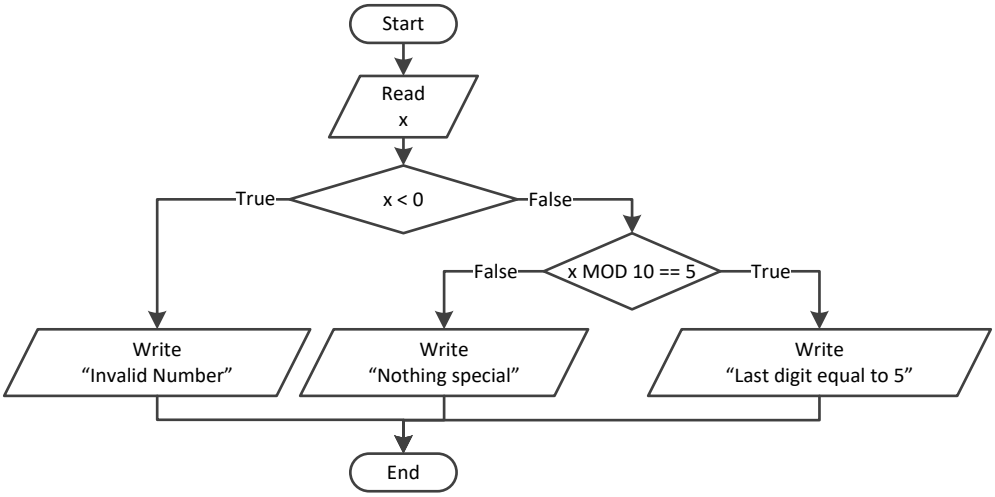
Chapter 22

22.7 Review Exercises

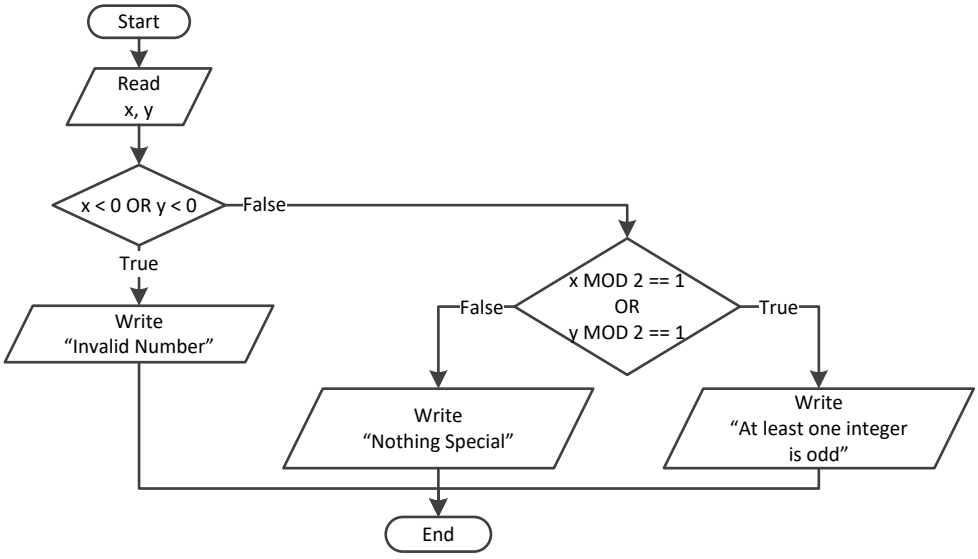
1. Solution

```
from math import sqrt

x = float(input("Enter a non-negative number: "))
if x < 0:
    print("Error! You entered a negative value")
else:
    print("The square root of", x, "is", sqrt(x))
```

2. Solution

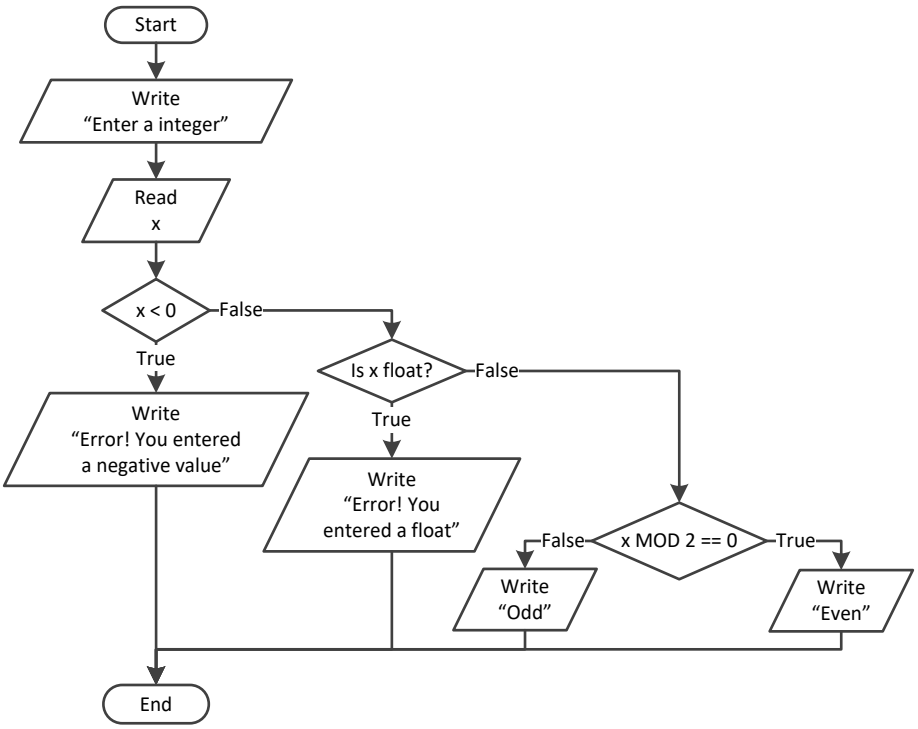


3. Solution



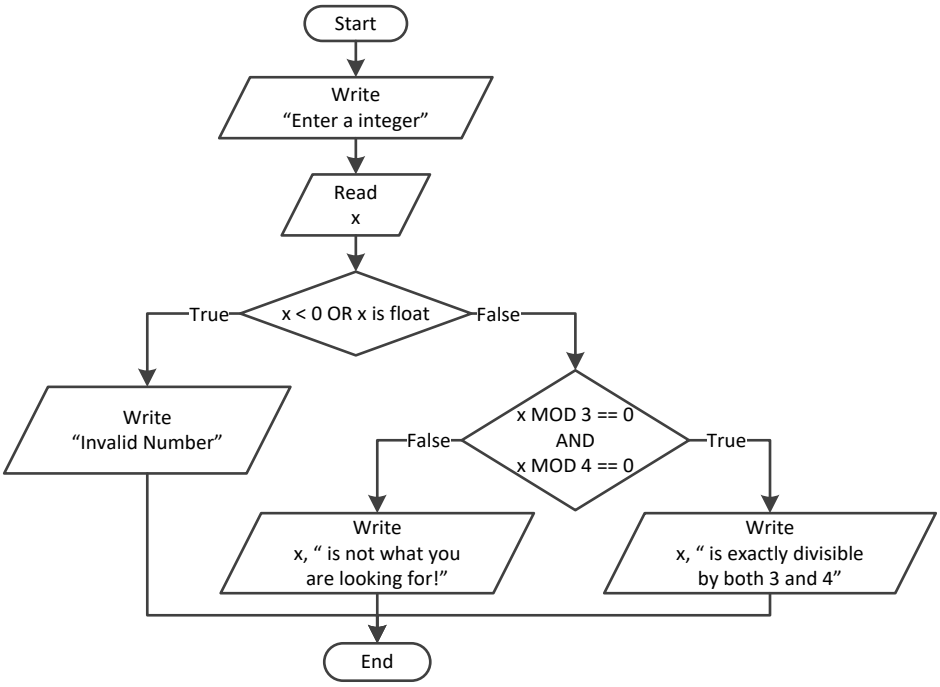
```
x = int(input())
y = int(input())
if x < 0 or y < 0:
    print("Invalid Number")
else:
    if x % 2 == 1 or y % 2 == 1:
        print("At least one integer is odd")
    else:
        print("Nothing Special")
```

4. Solution



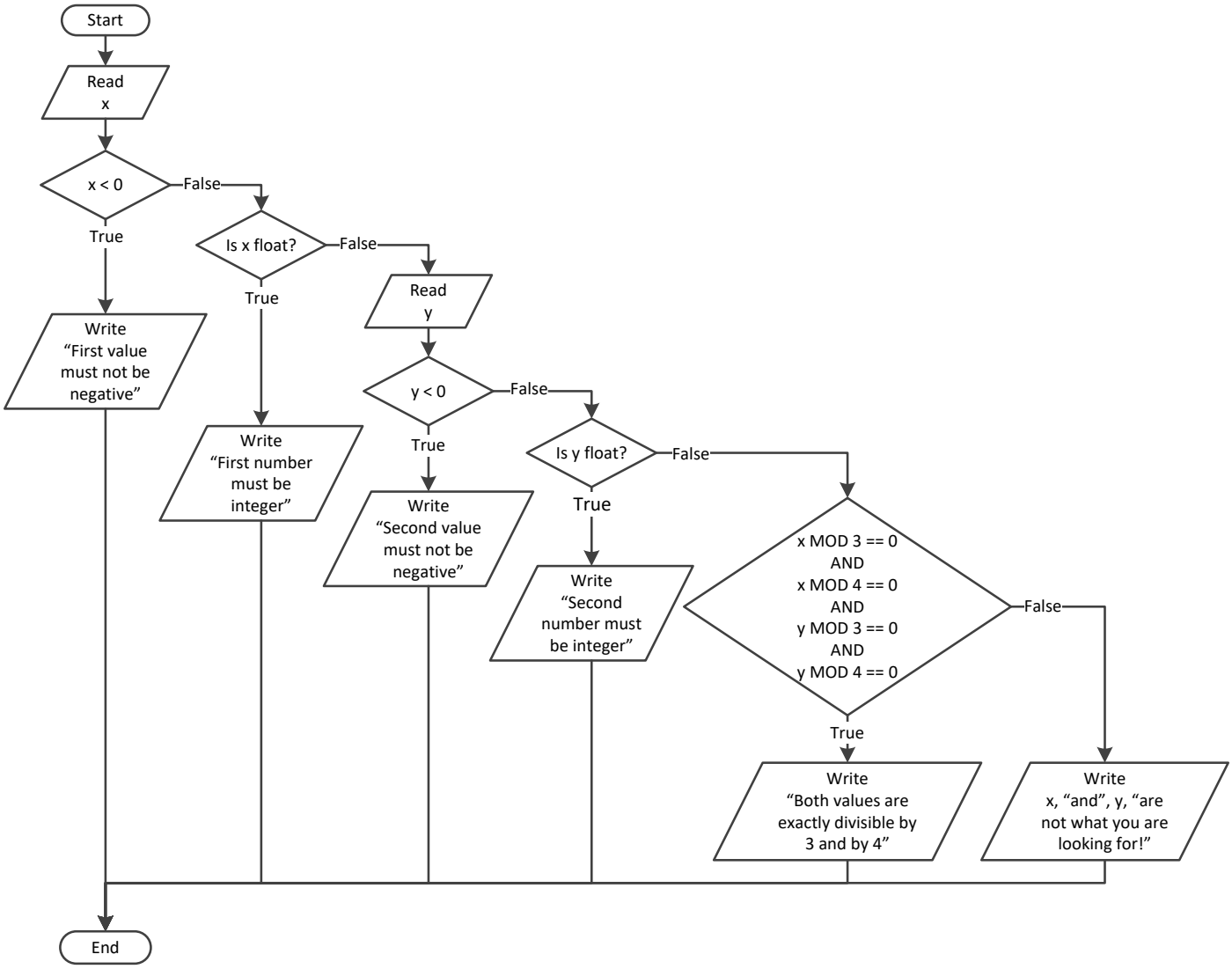

```
x = float(input("Enter a non-negative number: "))
if x < 0:
    print("Error! You entered a negative value")
elif x != int(x):
    print("Error! You entered a float")
elif x % 2 == 0:
    print("Even")
else:
    print("Odd")
```

5. Solution




```
x = float(input("Enter an integer: "))

if x < 0 or x != int(x):
    print("Invalid Number")
elif x % 3 == 0 and x % 4 == 0:
    print(x, "is exactly divisible by both 3 and 4")
else:
    print(x, "is not what you are looking for!")
```

6. Solution




```
x = float(input())
if x < 0:
    print("First value must be not be negative")
else:
    if x != int(x):
        print("First number must be integer")
    else:
        y = float(input())
        if y < 0:
            print("Second value must be not be negative")
        else:
            if y != int(y):
                print("Second number must be integer")
            else:
                if x % 3 == 0 and x % 4 == 0 and y % 3 == 0 and y % 4 == 0:
```

```
                print("Both values are exactly divisible by 3 and by 4")
            else:
                print("nothing Special")
```

7. Solution

```
print("1. Convert Kelvin to Fahrenheit")
print("2. Convert Fahrenheit to Kelvin")
print("3. Convert Fahrenheit to Celsius")
print("4. Convert Celsius to Fahrenheit")

choice = int(input("Enter a choice: "))
t = float(input("Enter a temperature: "))

if choice < 1 or choice > 4:
    print("Wrong choice")
elif choice == 1:
    if t < 0:  #Absolute zero in Kelvin
        print("Wrong temperature")
    else:
        print(1.8 * t - 459.67)
elif choice == 2:
    if t < -459.67:  #Absolute zero in Fahrenheit
        print("Wrong temperature")
    else:
        print((t + 459.57) / 1.8)
elif choice == 3:
    if t < -459.67:  #Absolute zero in Fahrenheit
        print("Wrong temperature")
    else:
        print(5 / 9 * (t - 32))
elif choice == 4:
    if t < -273.15:  #Absolute zero in Celcius
        print("Wrong temperature")
    else:
        print(9 / 5 * t + 32)
```

8. Solution

```
a = int(input("Enter 1st integer: "))
op = input("Enter type of operation: ").upper()
b = int(input("Enter 2nd integer: "))

message = "The result of " + str(a) + " " + op + " " + str(b) + " equals "

if op == "+":
    message += str(a + b)  #Concatenate
elif op == "-":
    message += str(a - b)  #Concatenate
elif op == "*":
```

```
    message += str(a * b)  #Concatenate
elif op == "/":
    if b == 0:
        message = "Infinite"  #Replace
    else:
        message += str(a / b)  #Concatenate
elif op == "DIV":
    if b == 0:
        message = "Infinite"  #Replace
    else:
        message += str(a // b)  #Concatenate
elif op == "MOD":
    if b == 0:
        message = "Infinite"  #Replace
    else:
        message += str(a % b)  #Concatenate
elif op == "POWER":
    message += str(a ** b)  #Concatenate

print(message)
```

9. Solution

```
a = int(input("Enter 1st integer: "))
op = input("Enter type of operation: ").upper()
b = int(input("Enter 2nd integer: "))

message = "The result of " + str(a) + " " + op + " " + str(b) + " equals "

if op == "+":
    message += str(a + b)  #Concatenate
elif op == "-":
    message += str(a - b)  #Concatenate
elif op == "*":
    message += str(a * b)  #Concatenate
elif op == "/":
    if b == 0:
        message = "Infinite"  #Replace
    else:
        message += str(a / b)  #Concatenate
elif op == "DIV":
    if b == 0:
        message = "Infinite"  #Replace
    else:
        message += str(a // b)  #Concatenate
elif op == "MOD":
    if b == 0:
        message = "Infinite"  #Replace
    else:
        message += str(a % b)  #Concatenate
```

```
elif op == "POWER":
    message += str(a ** b)  #Concatenate
else:
    message = "Error: Invalid operator"  #Replace

print(message)
```

10. Solution

```
a1 = int(input("Enter the age of the first person: "))
n1 = input("Enter the name of the first person: ")
a2 = int(input("Enter the age of the second person: "))
n2 = input("Enter the name of the second person: ")
a3 = int(input("Enter the age of the third person: "))
n3 = input("Enter the name of the third person: ")

minimum = a1
minName = n1
if a2 < minimum:
    minimum = a2
    minName = n2
if a3 < minimum:
    minimum = a3
    minName = n3

maximum = a1
maxName = n1
if a2 > maximum:
    maximum = a2
    maxName = n2
if a3 > maximum:
    maximum = a3
    maxName = n3

print(minName , maxName)
```

11. Solution

```
artistName = input("Enter artist's name: ")
score1 = int(input("Enter score No 1: "))
score2 = int(input("Enter score No 2: "))
score3 = int(input("Enter score No 3: "))
score4 = int(input("Enter score No 4: "))
score5 = int(input("Enter score No 5: "))

minimum = score1
if score2 < minimum:
    minimum = score2
if score3 < minimum:
    minimum = score3
```

```
if score4 < minimum:
    minimum = score4
if score5 < minimum:
    minimum = score5

maximum = score1
if score2 > maximum:
    minimum = score2
if score3 > maximum:
    minimum = score3
if score4 > maximum:
    minimum = score4
if score5 > maximum:
    minimum = score5

totalScore = score1 + score2 + score3 + score4 + score5 - minimum - maximum
print(artistName, "received", totalScore, "points")
```

12. Solution

```
age1 = int(input("Enter age for person No1:"))
age2 = int(input("Enter age for person No2:"))
age3 = int(input("Enter age for person No3:"))

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
print(middle)
```

13. Solution

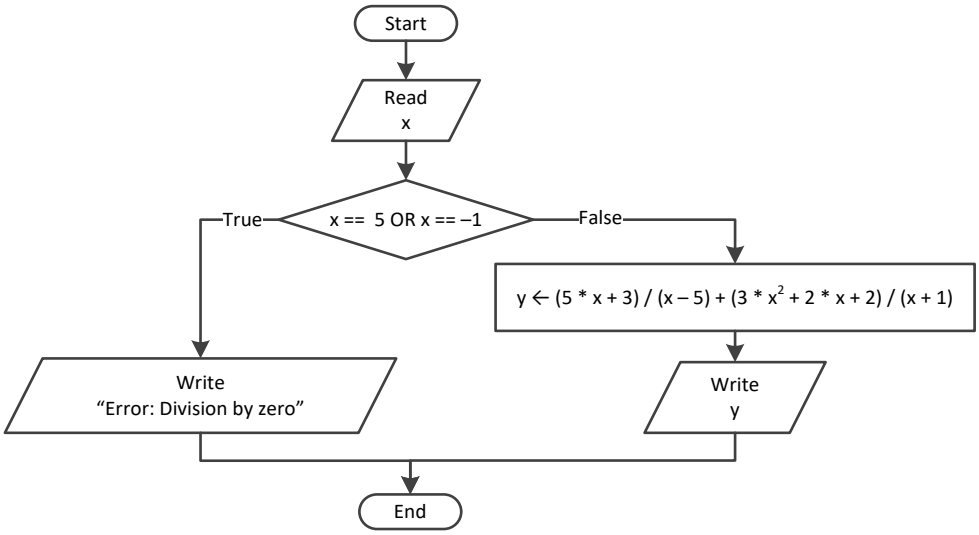
```
a1 = int(input("Enter the age of the first person: "))
n1 = input("Enter the name of the first person: ")
a2 = int(input("Enter the age of the second person: "))
n2 = input("Enter the name of the second person: ")
a3 = int(input("Enter the age of the third person: "))
n3 = input("Enter the name of the third person: ")

minimum = a1
minName = n1
if a2 < minimum:
```

```
    minimum = a2
    minName = n2
if a3 < minimum:
    minimum = a3
    minName = n3

maximum = a1
maxName = n1
if a2 > maximum:
    maximum = a2
    maxName = n2
if a3 > maximum:
    maximum = a3
    maxName = n3

middle = a1 + a2 + a3 - minimum - maximum

if abs(maximum - middle) < abs(minimum - middle):
    print(maxName)
else:
    print(minName)
```

14. Solution

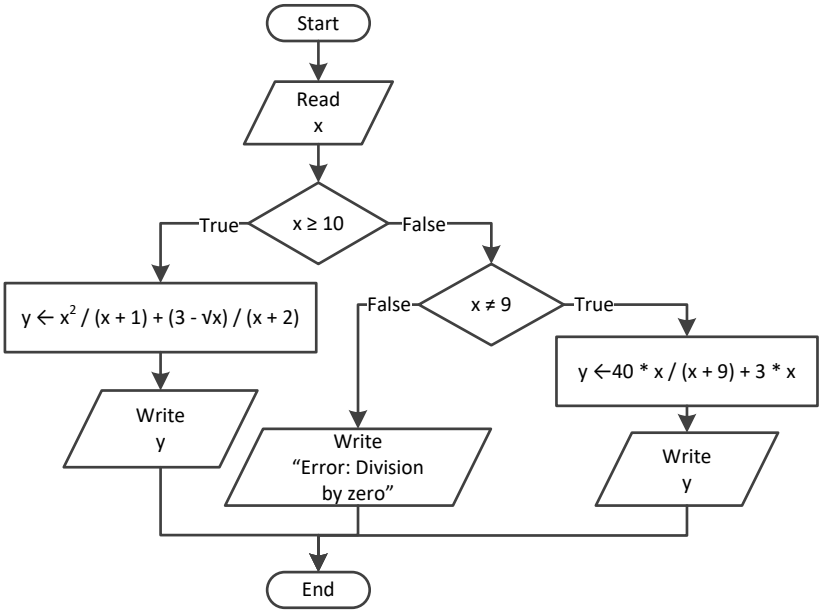
```
price1 = float(input())
title1 = input()
price2 = float(input())
title2 = input()
price3 = float(input())
title3 = input()

minimum = price1
minName = title1
if price2 < minimum:
    minimum = price2
    minName = title2
if price3 < minimum:
    minimum = price3
    minName = title3

amount = price1 + price2 + price3 - minimum
print("You need to pay: $", amount, sep = "")
print("Title provided for free:", minName)
print("You saved: $", minimum, sep = "")
```

15. Solution




```
x = float(input())

if x == 5 or x == -1:
    print("Error: Division by zero")
else:
    y = (5 * x + 3) / (x - 5) + (3 * x ** 2 + 2 * x + 2) / (x + 1)
    print(y)
```

16. Solution




```
from math import sqrt

x = float(input())
```

```
if x >= 10:
    y = x ** 2 / (x + 1) + (3 - sqrt(x)) / (x + 2)
    print(y)
elif x != 9:
    y = 40 * x / (x + 9) + 3 * x
    print(y)
else:
    print("Error: Division by zero")
```

17. Solution

```
x = float(input())

if x >= 0:
    y = 40 * x / (x - 5) + 3
    print(y)
elif x == 0 or x == 3:
    print("Error: Division by zero!")
else:
    y = (7 + x) / (x - 3) + (3 - x) / x
    print(y)
```

18. Solution

```
from math import sqrt

x = float(input())

if x <= -15 or x > 25:
    y = x - 1
    print(y)
elif x <= -10:
    y = x / sqrt(x + 30) + (8 + x) ** 2 / (x + 1)
    print(y)
elif x <= 0:
    y = abs(40 * x) / (x - 8)
    print(y)
else:
    if x == 9:
        print("Error: Division by zero")
    elif x < 9:
        print("Error: Invalid square root")
    else:
        y = 3 * x / sqrt(x - 9)
        print(y)
```

19. Solution

```
x = float(input("Enter a three-digit integer: "))

if x != int(x):
```

```
    print("Error! You must enter an integer")
elif x < 100 or x > 999:
    print("Entered integer is not a three-digit integer")
else:
    digit1, r = divmod(x, 100)
    digit2, digit3 = divmod(r, 10)

    total = digit1 ** 3 + digit2 ** 3 + digit3 ** 3

    if total == x:
        print("You entered an Armstrong number!")
    else:
        print("You entered a non-Armstrong number!")
```

20. Solution

```
d = int(input("Enter day 1 - 31: "))
m = int(input("Enter month 1 - 12: "))
y = int(input("Enter year: "))

if m == 2:
    if y % 4 == 0 and y % 100 != 0 or y % 400 == 0:
        print(29 - d)
    else:
        print(28 - d)
elif m == 4 or m == 6 or m == 9 or m == 11:
    print(30 - d)
else:
    print(31 - d)
```

21. Solution

```
word = input()

word1 = word[0].upper() +       \
        word[1].lower() +       \
        word[2].upper() +       \
        word[3].lower() +       \
        word[4].upper() +       \
        word[5].lower()

word2 = word[0].lower() +       \
        word[1].upper() +       \
        word[2].lower() +       \
        word[3].upper() +       \
        word[4].lower() +       \
        word[5].upper()

if word == word1 or word == word2:
    print("Word is okay!")
```

```
else:
    print("Word is not okay")
```

22. Solution

```
q = int(input("Enter quantity: "))

if q < 3:
    discount = 0
elif q < 6:
    discount = 10
elif q < 10:
    discount = 15
elif q < 14:
    discount = 20
elif q < 20:
    discount = 27
else:
    discount = 30

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

print("You got a discount of ", discount, "%", sep = "")
print("You must pay $", payment, sep = "")
```

23. Solution

```
VAT = 0.19

amount = float(input("Enter a before-tax amount: "))

if amount < 0:
    print("Error! You entered a negative value")
else:
    if amount < 50:
        discount = 0
    elif amount < 100:
        discount = 1
    elif amount < 250:
        discount = 2
    else:
        discount = 3

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

    print("You got a discount of ", discount, "%", sep ="")
    print("You must pay $", payment, sep = "")
```

24. Solution

```
a = int(input("Enter age: "))
if a < 18:
    print("Invalid age")
else:
    w = int(input("Enter weight in pounds: "))
    h = int(input("Enter height in inches: "))

    bmi = w * 703 / h ** 2

    if bmi < 15:
        print("Very severely underweight")
    elif bmi < 16:
        print("Severely underweight")
    elif bmi < 18.5:
        print("Underweight")
    elif bmi < 25:
        print("Normal")
    elif bmi < 30:
        print("Overweight")
    elif bmi < 35:
        print("Severely overweight")
    else:
        print("Very severely overweight")
```

25. Solution

```
TAX_RATE = 0.10

water = int(input("Enter water consumption (in cubic feet): "))

if water < 0:
    print("Error! You entered a negative value")
else:
    if water <= 10:
        total = water * 3
    elif water <= 20:
        total = 10 * 3 + (water - 10) * 5
    elif 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
    print("Total amount to pay (taxes included):", total)
```

26. Solution

```
income = float(input("Enter taxable income: "))
children = int(input("Enter number of children: "))

if income <= 8000:
    tax = income * 0.10
elif income <= 30000:
    tax = 8000 * 0.10 + (income - 8000) * 0.15
elif 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

print("Tax:", tax)
```

27. Solution

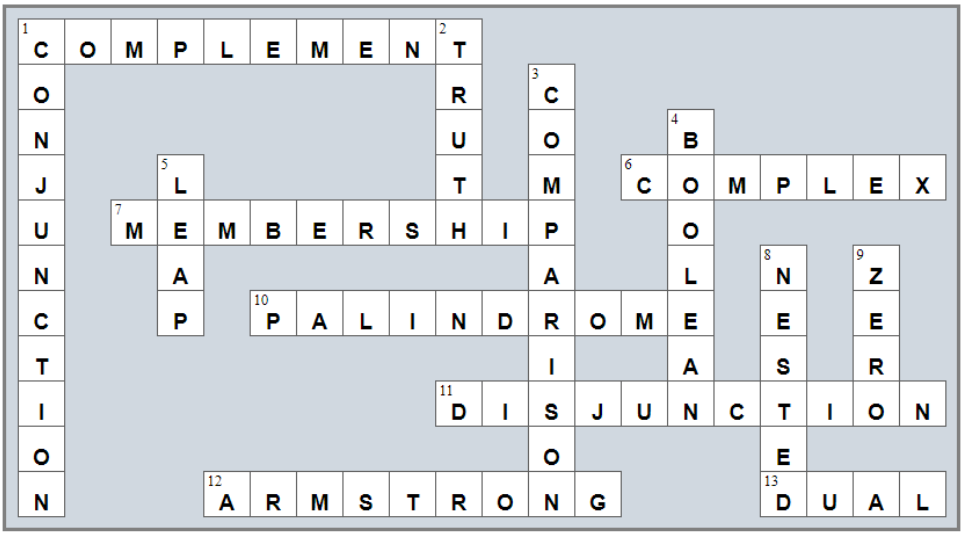
```
wind = float(input("Enter wind speed (in miles/hour): "))

if wind < 0:
    print("Error! You entered a negative value ")
else:
    if wind < 1:
        print("Beaufort: 0\nCalm")
    elif wind < 4:
        print("Beaufort: 1\nLight air")
    elif wind < 8:
        print("Beaufort: 2\nLight breeze")
    elif wind < 13:
        print("Beaufort: 3\nGentle breeze")
    elif wind < 18:
        print("Beaufort: 4\nModerate breeze")
    elif wind < 25:
        print("Beaufort: 5\nFresh breeze")
    elif wind < 31:
        print("Beaufort: 6\nStrong breeze")
    elif wind < 39:
        print("Beaufort: 7\nModerate gale")
    elif wind < 47:
        print("Beaufort: 8\nGale")
    elif wind < 55:
        print("Beaufort: 9\nStrong gale")
    elif wind < 64:
        print("Beaufort: 10\nStorm")
    elif wind < 74:
```

```
        print("Beaufort: 11\nViolent storm")
    else:
        print("Beaufort: 12\nHurricane force")

    if wind < 13:
        print("It's Fishing Day!!!")
```

Review in “Decision Control Structures”

Review Crossword Puzzle

1.



Chapter 23

23.3 Review Questions: True/False

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

Chapter 24

24.4 Review Questions: True/False

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

24.5 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

24.6 Review Exercises

1. Solution

```
i = 3
while True:
    print(i)
    i -= 1
    if i <= 0: break  #Alternatively you can use the logical operator ==
print("The end")
```

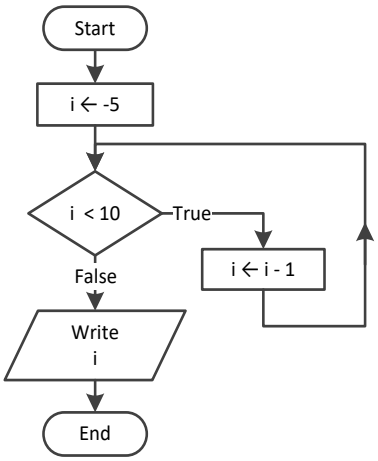
2. Solution

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

10	`i -= 1`	**0**	3
11	`x += i`	0	**3**
12	`while i >= 0:`	True	
13	`i -= 1`	**-1**	3
14	`x += i`	-1	**2**
15	`while i >= 0:`	False	
16	`print(x)`	It displays: 2	

It performs 4 iterations

3. Solution



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

It performs an infinite number of iterations

4. Solution

Step	Statement	a	b	c	d
1	`a = 2`	**2**	?	?	?
2	`while a <= 10:`	True			

3	`b = a + 1`	2	**3**	?	?
4	`c = b * 2`	2	3	**6**	?
5	`d = c - b + 1`	2	3	6	**4**
6	`if d == 4:`	True			
7	`print(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	`if d == 4:`	False			
14	`elif d == 5:`	False			
15	`elif d == 8:`	True			
16	`print(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	`if d == 4:`	False			
23	`elif d == 5:`	False			
24	`elif d == 8:`	False			
25	`print(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 += 1`	**39**	10
5	`print(y)`	It displays: 10	
6	`if y >= x: break`	False	
7	`y *= 2`	39	**20**
8	`x += 1`	**40**	20
9	`print(y)`	It displays: 20	
10	`if y >= x: break`	False	
11	`y *= 2`	40	**40**
12	`x += 1`	**41**	40
13	`print(y)`	It displays: 40	
14	`if y >= x: break`	False	
15	`y *= 2`	41	**80**

16	`x += 1`	**42**	80
17	`print(y)`	It displays: 80	
18	`if y >= x: break`	True	

8. Solution

Step	Statement	Notes	x
1	`x = 1`		**1**
2	`if x % 2 == 0:`	False	
3	`x += 3`		**4**
4	`print(x)`	It displays: 4	
5	`if x >= 12: break`	False	
6	`if x % 2 == 0:`	True	
7	`x += 1`		**5**
8	`print(x)`	It displays: 5	
9	`if x >= 12: break`	False	
10	`if x % 2 == 0:`	False	
11	`x += 3`		**8**
12	`print(x)`	It displays: 8	
13	`if x >= 12: break`	False	
14	`if x % 2 == 0:`	True	
15	`x += 1`		**9**
16	`print(x)`	It displays: 9	
17	`if x >= 12: break`	False	
18	`if x % 2 == 0:`	False	
19	`x += 3`		**12**
20	`print(x)`	It displays: 12	
21	`if x >= 12: break`	True	

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	`print(x, ",", y)`	It displays: 4, 4	

7	`if y >= 65535: break`	False	
8	`y = y ** 2`	4	**16**
9	`if x < 256:`	True	
10	`x = x + y`	**20**	16
11	`print(x, ",", y)`	It displays: 20, 16	
12	`if y >= 65535: break`	False	
13	`y = y ** 2`	20	**256**
14	`if x < 256:`	True	
15	`x = x + y`	**276**	256
16	`print(x, ",", y)`	It displays: 276, 256	
17	`if y >= 65535: break`	False	
18	`y = y ** 2`	276	**65536**
19	`if x < 256:`	False	
20	`print(x, ",", y)`	It displays: 276, 65536	
21	`if y >= 65535: break`	True	

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	`elif 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	`if c >= 11: break`	False				
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	`if c >= 11: break`			False		
18	`x = a + b`	5	5	5	5	**10**
19	`if x % 2 != 0:`			False		
20	`elif 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	`if c >= 11: break`			False		
25	`x = a + b`	5	5	8	5	**10**
26	`if x % 2 != 0:`			False		
27	`elif 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	`if c >= 11: break`			True		

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

```
n = int(input())
total = 0

i = 1
```

```
while i <= n:
    a = float(input())
    total = total + a
    i += 1

print(total)
if n > 0:
    print(total / n)
```

14. Solution

```
count = 0

n = int(input())
p = 1

i = 1
while i <= n:
    a = int(input())
    if a % 2 == 0:
        p = p * a
        count += 1
    i += 1

if count > 0:
    print(p)
else:
    print("You entered no even integers")
```

15. Solution

```
total = 0

i = 1
while i <= 100:
    a = int(input())
    if a % 10 == 0:
        total = total + a
    i += 1
print(total)
```

16. Solution

```
total = 0

i = 1
while i <= 20:
    a = int(input())
    if 100 <= a <= 999:
        total = total + a
```

```
    i += 1
print(total)
```

17. Solution

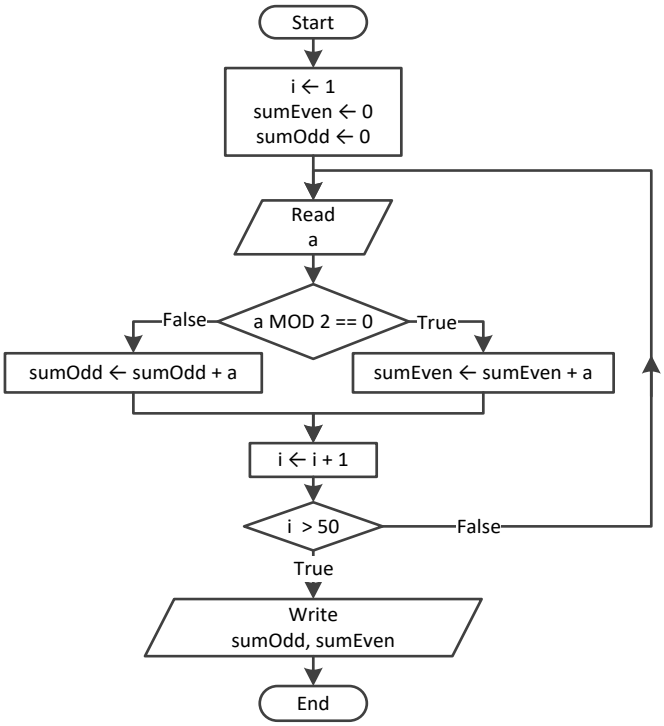
```
p = 1

a = float(input())
while a != 0:
    p = p * a
    a = float(input())
print(p)
```

Step	Statement	a	p
1	p = 1	?	**1**
2	a = float(input())	**3.0**	1
3	while a != 0:	True	
4	p = p * a	3.0	**3**
5	a = float(input())	**2.0**	3
6	while a != 0:	True	
7	p = p * a	2.0	**6**
8	a = float(input())	**9.0**	6
9	while a != 0:	True	
10	p = p * a	9.0	**54**
11	a = float(input())	**0.0**	54
12	while a != 0:	False	
13	print(p)	It displays: 54	

18. Solution

```
population = 30000

years = 0
while population <= 100000:
    population += population * 0.03
    years += 1

print(years)
```

19. Solution




```
i = 1
sumEven = 0
sumOdd = 0
while True:
    a = int(input())
    if a % 2 == 0:
        sumEven += a
    else:
        sumOdd += a
    i += 1
    if i > 50: break
print(sumOdd, sumEven)
```

20. Solution

```
n = int(input())
i = 1
p = 1
while True:
    a = int(input())
    if a < 0:
        p *= a
    i += 1
    if i > n: break
print(abs(p))
```

21. Solution

```
i = 1
p = 1
while True:
    a = int(input("Enter an integer: "))
    if 500 <= a <= 599:
        p *= a
    i += 1
    if i > 5: break

print(p)
```

22. Solution

```
population = 50000

years = 0
while True:
    population -= population * 0.10
    years += 1
    if population < 20000: break

print(years)
```

Chapter 25

25.3 Review Questions: True/False

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

25.4 Review Questions: Multiple Choice

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

25.5 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	`if j < 5:`		True	
5	`b += 1`	0	**1**	0
6	`j = 2`	0	1	**2**
7	`if j < 5:`		True	
8	`b += 1`	0	**2**	2
9	`j = 4`	0	2	**4**
10	`if j < 5:`		True	
11	`b += 1`	0	**3**	4
12	`j = 6`	0	3	**6**
13	`if j < 5:`		False	
14	`a += j - 1`	**5**	3	6
15	`j = 8`	5	3	**8**
16	`if j < 5:`		False	
17	`a += j - 1`	**12**	3	8

18	`print(a, ",", b)`	It displays: 12, 3

2. Solution

For input value of 10

Step	Statement	a	b	j
1	`a = int(input())`	**10**	?	?
2	`b = a`	10	**10**	?
3	`j = a - 5`	10	10	**5**
4	`if j % 2 != 0:`		True	
5	`b = a + j + 5`	10	**20**	5
6	`j = 7`	10	20	**7**
7	`if j % 2 != 0:`		True	
8	`b = a + j + 5`	10	**22**	7
9	`j = 9`	10	22	**9**
10	`if j % 2 != 0:`		True	
11	`b = a + j + 5`	10	**24**	9
12	`print(b)`	It displays: 24		

For input value of 21

Step	Statement	a	b	j
1	`a = int(input())`	**21**	?	?
2	`b = a`	21	**21**	?
3	`j = a - 5`	21	21	**16**
4	`if j % 2 != 0:`		False	
5	`b = a - j`	21	**5**	16
6	`j = 18`	21	5	**18**
7	`if j % 2 != 0:`		False	
8	`b = a - j`	21	**3**	18
9	`j = 20`	21	3	**20**
10	`if j % 2 != 0:`		False	
11	`b = a - j`	21	**1**	20
12	`print(b)`	It displays: 1		

3. Solution

For input value of 12

Step	Statement	a	x	y	j
1	`a = int(input())`	**12**	?	?	?

2	`j = 2`	12	?	?	**2**
3	`x = j * 3 + 3`	12	**9**	?	2
4	`y = j * 2 + 10`	12	9	**14**	2
5	`if y - x > 0 or x > 30:`	True			
6	`y *= 2`	12	9	**28**	2
7	`x += 4`	12	**13**	28	2
8	`print(x, ",", y)`	It displays: 13, 28			
9	`j = 5`	12	13	28	**5**
10	`x = j * 3 + 3`	12	**18**	28	5
11	`y = j * 2 + 10`	12	18	**20**	5
12	`if y - x > 0 or x > 30:`	True			
13	`y *= 2`	12	18	**40**	5
14	`x += 4`	12	**22**	40	5
15	`print(x, ",", y)`	It displays: 22, 40			
16	`j = 8`	12	22	40	**8**
17	`x = j * 3 + 3`	12	**27**	40	8
18	`y = j * 2 + 10`	12	27	**26**	8
19	`if y - x > 0 or x > 30:`	False			
20	`x += 4`	12	**31**	26	8
21	`print(x, ",", y)`	It displays: 31, 26			
22	`j = 11`	12	31	26	**11**
23	`x = j * 3 + 3`	12	**36**	26	11
24	`y = j * 2 + 10`	12	36	**32**	11
25	`if y - x > 0 or x > 30:`	True			
26	`y *= 2`	12	36	**64**	11
27	`x += 4`	12	**40**	64	11
28	`print(x, ",", y)`	It displays: 40, 64			

4. Solution

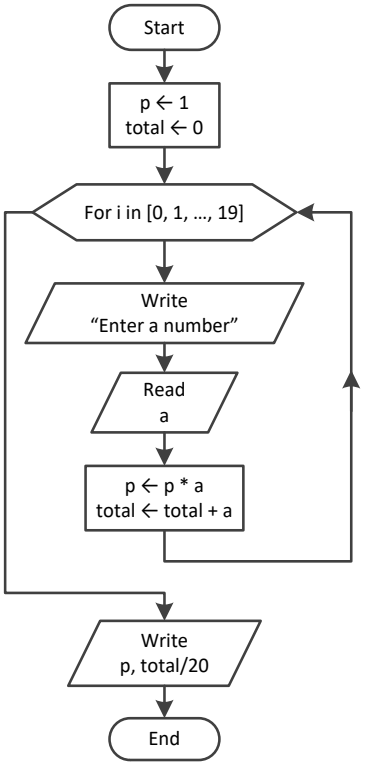
i. 9
ii. Any value greater than or equal to 20 and less than 25 ($20 \le x < 25$)
iii. −7 (or −6)
iv. −1

5. Solution

It displays:
Z

Zee
Zeeuuu
Zeeuuussss

6. Solution




```
p = 1
total = 0
for i in range(20):
    a = float(input("Enter a number: "))
    p = p * a
    total = total + a
print(p, total / 20)
```

7. Solution

```
from math import sin, pi

i = 0
while i <= 360:
    print(sin(i * pi / 180))
    i += 0.5
```

8. Solution

```
from math import cos, pi

deg = int(input("Enter degrees: "))
```

```
for i in range(0, deg + 1):
    print(cos(i * pi / 180))
```

9. Solution

```
s = 0
for i in range(1, 101, 2):
    s += i
print(s)
```

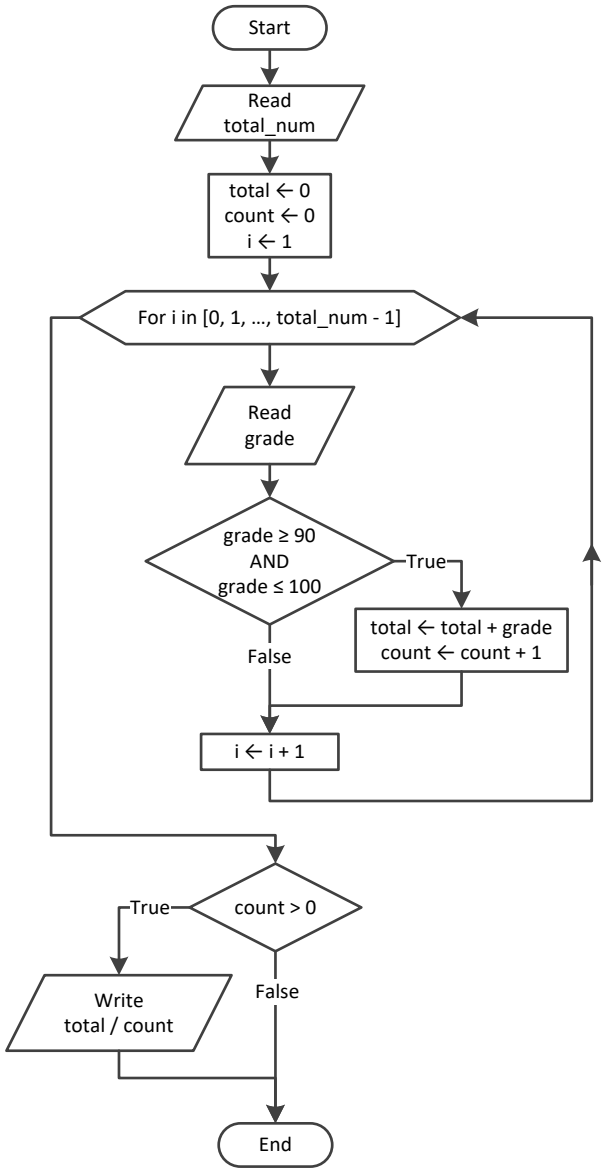
10. Solution

```
n = int(input())
p = 1
for i in range(2, 2 * n + 2, 2):
    p *= i ** (i - 1)
print(p)
```

11. Solution

```
s = 0
i = 1
offset = 0
while i <= 191:
    s += i
    offset += 1
    i += offset
print(s)
```

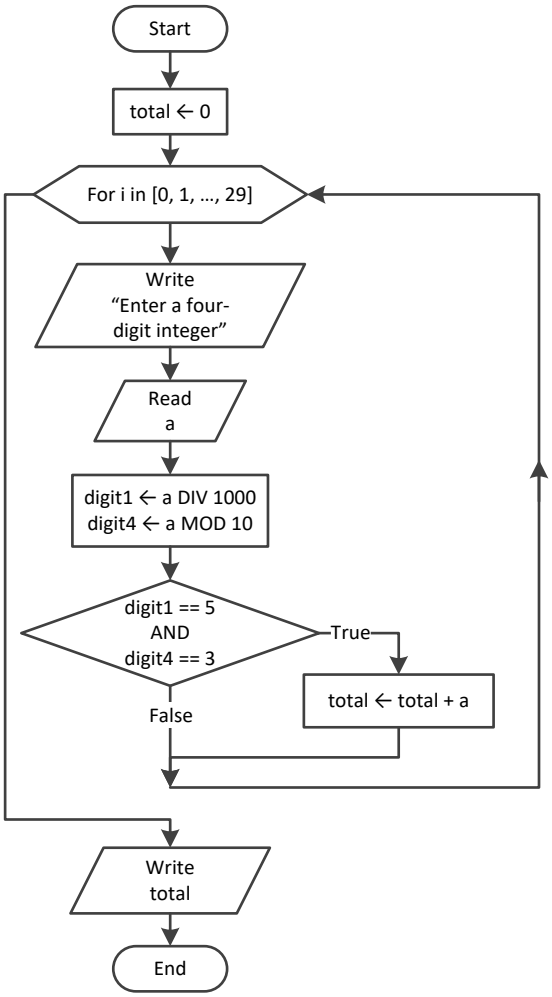
12. Solution




```
totalNum = int(input())
total = 0
count = 0
for i in range(totalNum):
    grade = int(input())
    if 90 <= grade <= 100:
        total += grade
        count += 1
if count > 0:
    print(total / count)
```

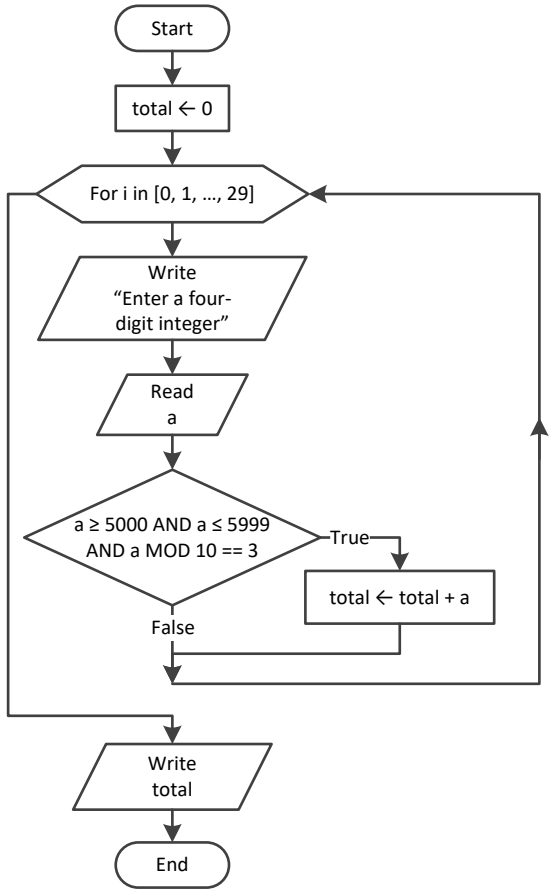
13. Solution

First approach



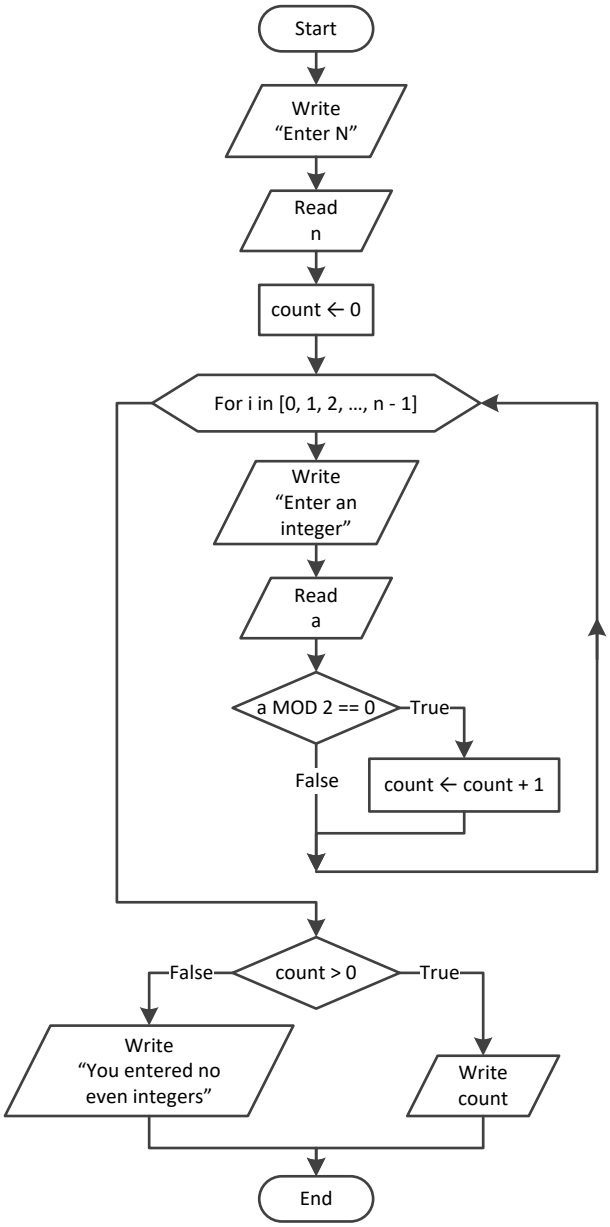

```
total = 0
for i in range(30):
    a = int(input("Enter a four-digit integer: "))
    digit1 = a // 1000
    digit4 = a % 10
    if digit1 == 5 and digit4 == 3:
        total += a

print(total)
```

Second approach



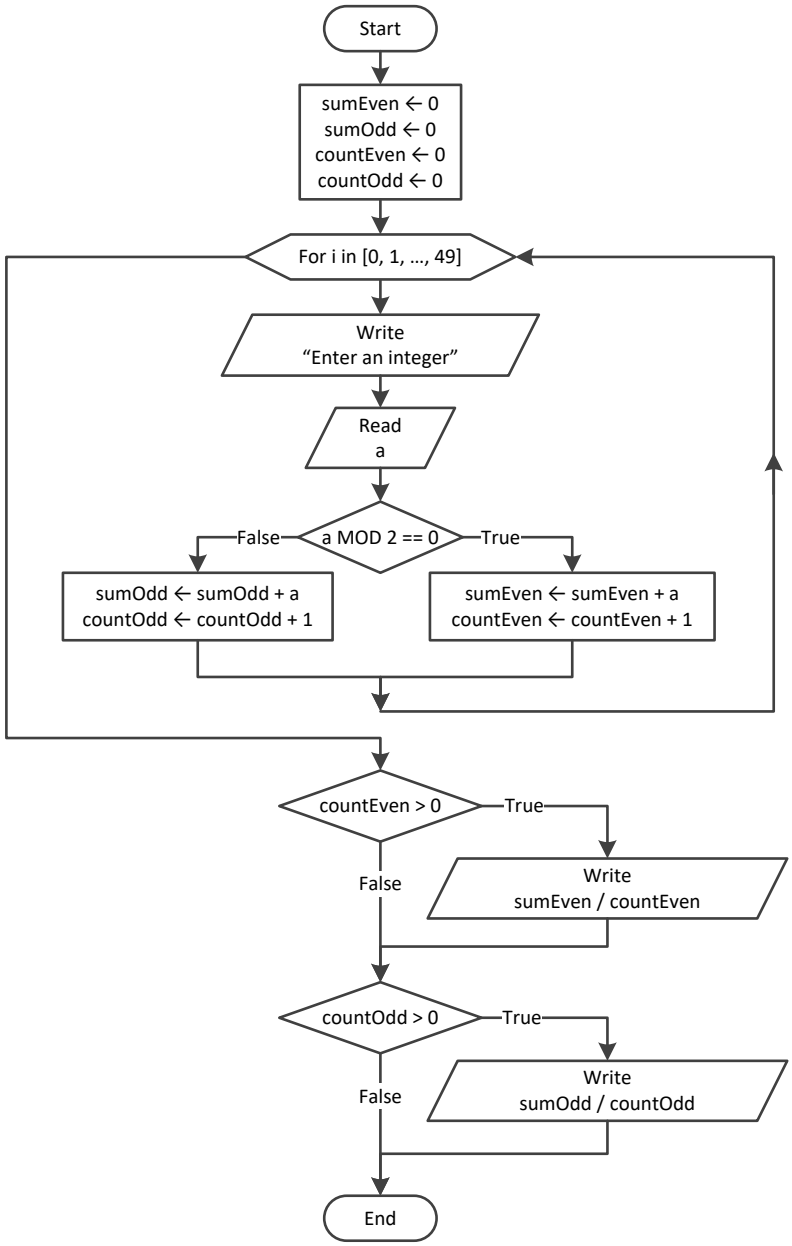

```
total = 0
for i in range(30):
    a = int(input("Enter a four-digit integer: "))
    if a >= 5000 and a <=5999 and a % 10 == 3:
        total += a

print(total)
```

14. Solution




```
n = int(input("Enter N: "))
count = 0
for i in range(n):
    a = int(input("Enter an integer: "))
    if a % 2 == 0:
        count += 1

if count > 0:
    print(count)
else:
    print("You entered no even integers")
```

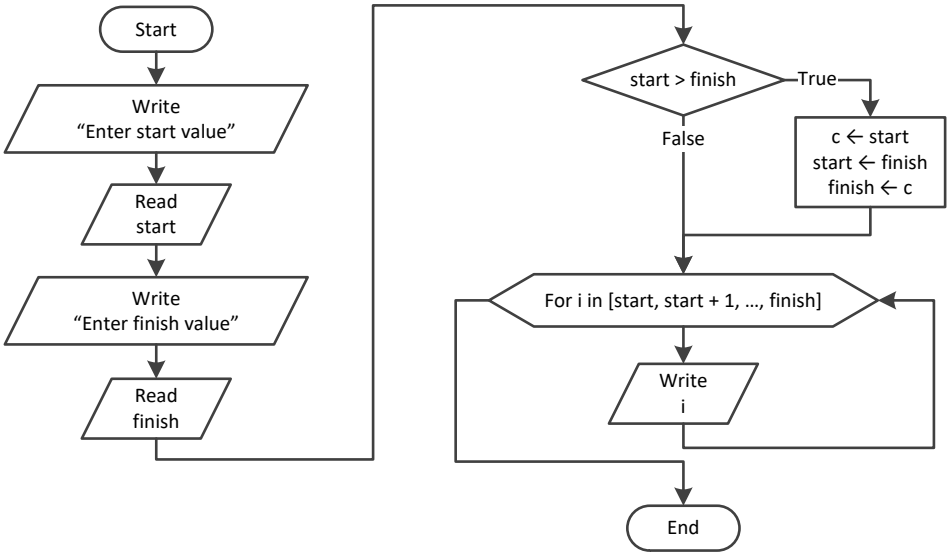
15. Solution




```
sumEven = 0
sumOdd = 0
countEven = 0
countOdd = 0
for i in range(50):
    a = int(input("Enter an integer: "))
    if a % 2 == 0:
        sumEven += a
        countEven += 1
    else:
        sumOdd += a
```

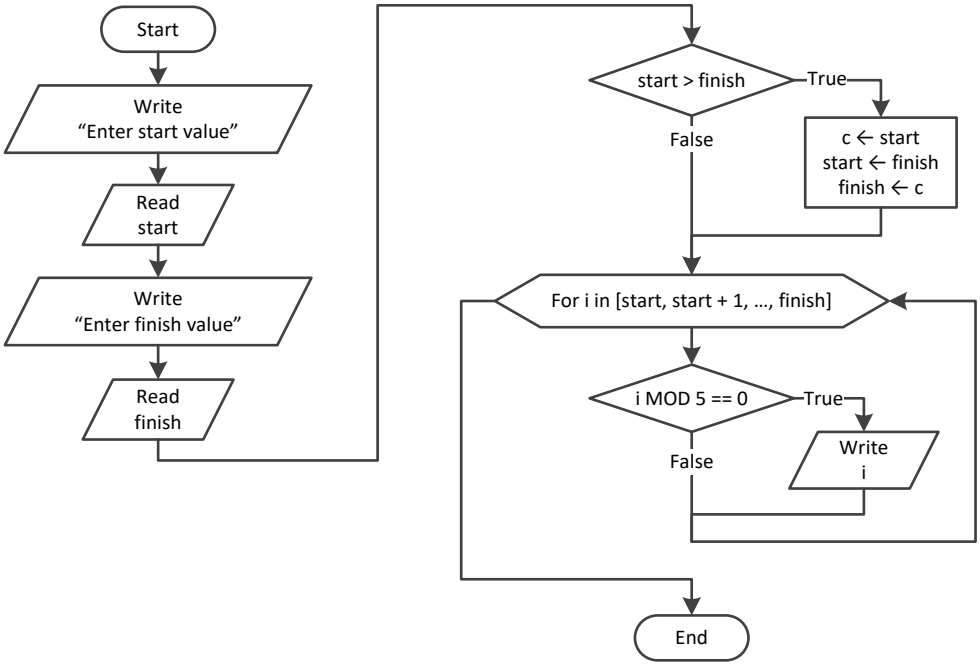
```
        countOdd += 1

if countEven > 0:
    print(sumEven / countEven)

if countOdd > 0:
    print(sumOdd / countOdd)
```

16. Solution




```
start = int(input("Enter start value: "))
finish = int(input("Enter finish value: "))

if start > finish:          # Or you can do the following:
    c = start               # start, finish = finish, start
    start = finish
    finish = c

for i in range(start, finish + 1):
    print(i)
```

17. Solution




```
start = int(input("Enter start value: "))
finish = int(input("Enter finish value: "))

if start > finish:
    c = start
    start = finish
    finish = c

for i in range(start, finish + 1):
    if i % 5 == 0:
        print(i)
```

18. Solution

First approach

```
b = float(input("Enter a value for base: "))
exp = int(input("Enter an integer for exponent: "))

p = 1
if exp >= 0:
    for i in range(exp):
        p *= b
else:
    for i in range(-exp):
        p *= 1 / b

print(p)
```

Second approach

```
b = float(input("Enter a value for base: "))
exp = int(input("Enter an integer for exponent: "))

p = 1
for i in range(abs(exp)):
    p *= b

if exp < 0:
    p = 1 / p

print(p)
```

19. Solution

```
msg = input("Enter a message: ")

characters = len(msg)
count = 0
for i in range(characters):
    if msg[i] == " ":
        count += 1

words = count + 1

print("The message entered contains", words, "words")
```

20. Solution

```
msg = input("Enter a message: ")

characters = len(msg)
count = 0
for i in range(characters):
    if msg[i] == " ":
        count += 1

words = count + 1

print("The average number of letters in each word is", (characters - count) / words)
```

21. Solution

```
consonants = "BCDFGHJKLMNPQRSTVWZYX"

message = input("Enter an English message: ")

count = 0
for character in message:
    if character.upper() in consonants:
```

```
        count += 1

print("Consonants:", count)
```

22. Solution

```
vowels = "AEIOU"
consonants = "BCDFGHJKLMNPQRSTVWZYX"
digits = "0123456789"

message = input("Enter an English message: ")

countv = countc = countd = 0
for character in message:
    if character.upper() in vowels:
        countv += 1
    elif character.upper() in consonants:
        countc += 1
    elif character in digits:  #No need to use the upper() method for digits
        countd += 1

print("Vowels:", countv)
print("Consonants:", countc)
print("Digits:", countd)
```

Chapter 26

26.3 Review Questions: True/False

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

26.4 Review Questions: Multiple Choice

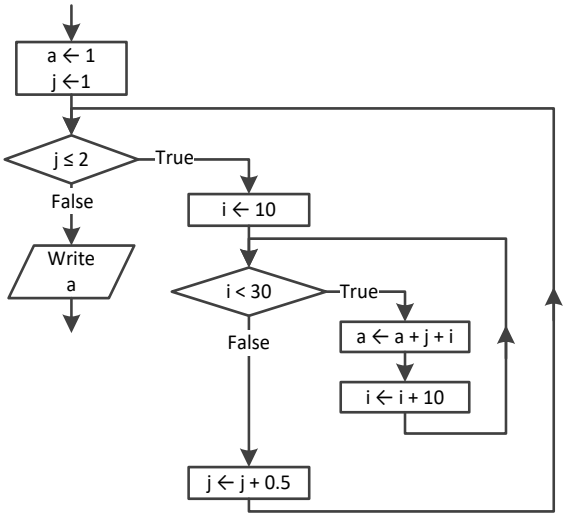
1. b
2. a
3. c
4. a
5. b

26.5 Review Exercises

1. Solution

i. 10
ii. A value greater than or equal to 45 and less than 50 ($45 \le x < 50$)
iii. −7 (or −8)
iv. 138 (or 139)

2. Solution



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

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

3. Solution

Step	Statement	s	i	j
1	`s = 0`	**0**	?	?
2	`i = 1`	0	**1**	?
3	`j = 3`	0	1	**3**

4	`s = s + i * j`	**3**	1	3
5	`j = 2`	3	1	**2**
6	`s = s + i * j`	**5**	1	2
7	`j = 1`	5	1	**1**
8	`s = s + i * j`	**6**	1	1
9	`i = 2`	6	**2**	1
10	`j = 3`	6	2	**3**
11	`s = s + i * j`	**12**	2	3
12	`j = 2`	12	2	**2**
13	`s = s + i * j`	**16**	2	2
14	`i = 3`	16	**3**	2
15	`j = 3`	16	3	**3**
16	`s = s + i * j`	**25**	3	3
17	`i = 4`	25	**4**	3
18	`print(s)`	It displays: 25		

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

4. Solution

For input value of "NO"

Step	Statement	s	y	i	ans
1	`s = 1`	**1**	?	?	?
2	`y = 25`	1	**25**	?	?
3	`i = 1`	1	25	**1**	?
4	`s = s + y`	**26**	25	1	?
5	`y -= 5`	26	**20**	1	?
6	`i = 2`	26	20	**2**	?
7	`s = s + y`	**46**	20	2	?
8	`y -= 5`	46	**15**	2	?
9	`i = 3`	46	15	**3**	?
10	`s = s + y`	**61**	15	3	?
11	`y -= 5`	61	**10**	3	?
12	`ans = input()`	61	10	3	**"NO"**
13	`if ans != "YES": break`	True			
14	`print(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	`s = s + y`	**26**	25	1	?
5	`y -= 5`	26	**20**	1	?
6	`i = 2`	26	20	**2**	?
7	`s = s + y`	**46**	20	2	?
8	`y -= 5`	46	**15**	2	?
9	`i = 3`	46	15	**3**	?
10	`s = s + y`	**61**	15	3	?
11	`y -= 5`	61	**10**	3	?
12	`ans = input()`	61	10	3	**“YES”**
13	`if ans != "YES": break`	False			
14	`i = 1`	61	10	**1**	“YES”
15	`s = s + y`	**71**	10	1	“YES”
16	`y -= 5`	71	**5**	1	“YES”
17	`i = 2`	71	5	**2**	“YES”
18	`s = s + y`	**76**	5	2	“YES”
19	`y -= 5`	76	**0**	2	“YES”
20	`i = 3`	76	0	**3**	“YES”
21	`s = s + y`	**76**	0	3	“YES”
22	`y -= 5`	76	**–5**	3	“YES”
23	`ans = input()`	76	–5	3	**“NO”**
24	`if ans != "YES": break`	True			
25	`print(s)`	It displays: 76			

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

Step	Statement	s	y	i	ans
1	`s = 1`	**1**	?	?	?
2	`y = 25`	1	**25**	?	?
3	`i = 1`	1	25	**1**	?
4	`s = s + y`	**26**	25	1	?
5	`y -= 5`	26	**20**	1	?
6	`i = 2`	26	20	**2**	?
7	`s = s + y`	**46**	20	2	?

8	`y -= 5`	46	**15**	2	?
9	`i = 3`	46	15	**3**	?
10	`s = s + y`	**61**	15	3	?
11	`y -= 5`	61	**10**	3	?
12	`ans = input()`	61	10	3	**"YES"**
13	`if ans != "YES": break`	False			
14	`i = 1`	61	10	**1**	"YES"
15	`s = s + y`	**71**	10	1	"YES"
16	`y -= 5`	71	**5**	1	"YES"
17	`i = 2`	71	5	**2**	"YES"
18	`s = s + y`	**76**	5	2	"YES"
19	`y -= 5`	76	**0**	2	"YES"
20	`i = 3`	76	0	**3**	"YES"
21	`s = s + y`	**76**	0	3	"YES"
22	`y -= 5`	76	**–5**	3	"YES"
23	`ans = input()`	76	–5	3	**"YES"**
24	`if ans != "YES": break`	False			
25	`i = 1`	76	–5	**1**	"YES"
26	`s = s + y`	**71**	–5	1	"YES"
27	`y -= 5`	71	**–10**	1	"YES"
28	`i = 2`	71	–10	**2**	"YES"
29	`s = s + y`	**61**	–10	2	"YES"
30	`y -= 5`	61	**–15**	2	"YES"
31	`i = 3`	61	–15	**3**	"YES"
32	`s = s + y`	**46**	–15	3	"YES"
33	`y -= 5`	46	**–20**	3	"YES"
34	`ans = input()`	46	–20	3	**"NO"**
35	`if ans != "YES": break`	True			
36	`print(s)`	It displays: 46			

5. Solution

```
for hour in range(24):
    for minutes in range(60):
        print(hour, "\t", minutes)
```

6. Solution

```
for i in range(5, 0, -1):
```

```
    for j in range(i):
        print(i, end = "")
    print()
```

7. Solution

```
for i in range(6):
    for j in range(i + 1):
        print(j, end = "")
    print()
```

8. Solution

First approach - Using nested loop control structures

```
for i in range(4):
    for j in range(10):
        print("* ", end = "")
    print()
```

Second approach - Without using any loop control structures (The amateur way!!!)

```
print("*  *  *  *  *  *  *  *  *  *")
print("*  *  *  *  *  *  *  *  *  *")
print("*  *  *  *  *  *  *  *  *  *")
print("*  *  *  *  *  *  *  *  *  *")
```

Third approach - Without using any loop control structures (The Pythonic way!!!)

```
print("*  " * 10)
print("*  " * 10)
print("*  " * 10)
print("*  " * 10)
```

Fourth approach - Without using any loop control structures (The Pythonic way No 2!!!)

```
print(("*  " * 10) + "\n") * 4)
```

9. Solution

First approach - Using nested loop control structures

```
y = int(input("Enter an integer between 3 and 20: "))

for i in range(y):
    for j in range(y):
        print("* ", end = "")
    print()
```

Second approach - Using one single loop control structure

```
y = int(input("Enter an integer between 3 and 20: "))

for i in range(y):
    print("*  " * y)
```

Third approach - Without using any loop control structures (The Pythonic way!!!)

```
y = int(input("Enter an integer between 3 and 20: "))
```

```
print(("*  " * y) + "\n") * y)
```

10. Solution

First approach - Using nested loop control structures

```
y = int(input("Enter an integer between 3 and 20: "))

for j in range(y):
    print("* ", end = "")
print()

for i in range(y - 2):
    print("* ", end = "")
    for j in range(y - 2):
        print("  ", end = "")
    print("* ")

for j in range(y):
    print("* ", end = "")
```

Second approach - Without using any loop control structures (The Pythonic way!!!)

```
y = int(input("Enter an integer between 3 and 20: "))

print(("* " * y) + "\n" + (("* " + "  " * (y - 2)) + "* \n") * (y - 2) + ("* " * y))
```

11. Solution

First approach - Using nested loop control structures

```
for i in range(1, 5):
    for j in range(i):
        print("* ", end = "")
    print()

for i in range(5, 0, -1):
    for j in range(i):
        print("* ", end = "")
    print()
```

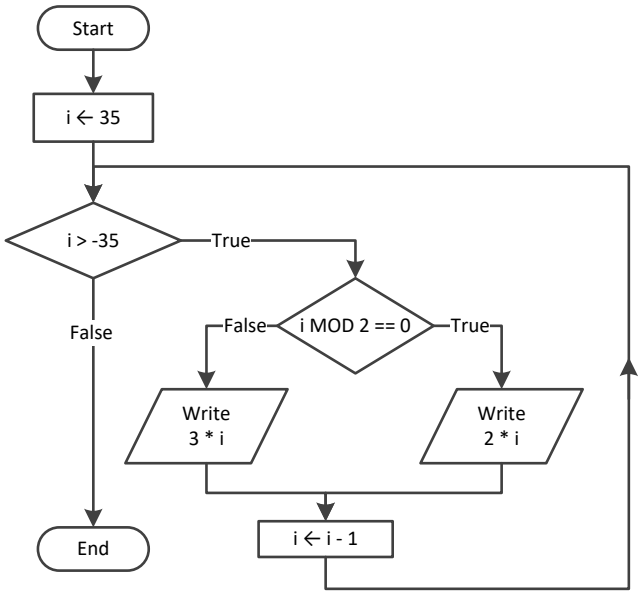
Second approach - Using just one for-loop (The Pythonic way!!!)

```
for i in [1, 2, 3, 4, 5, 4, 3, 2, 1]:
    print("* " * i)
```

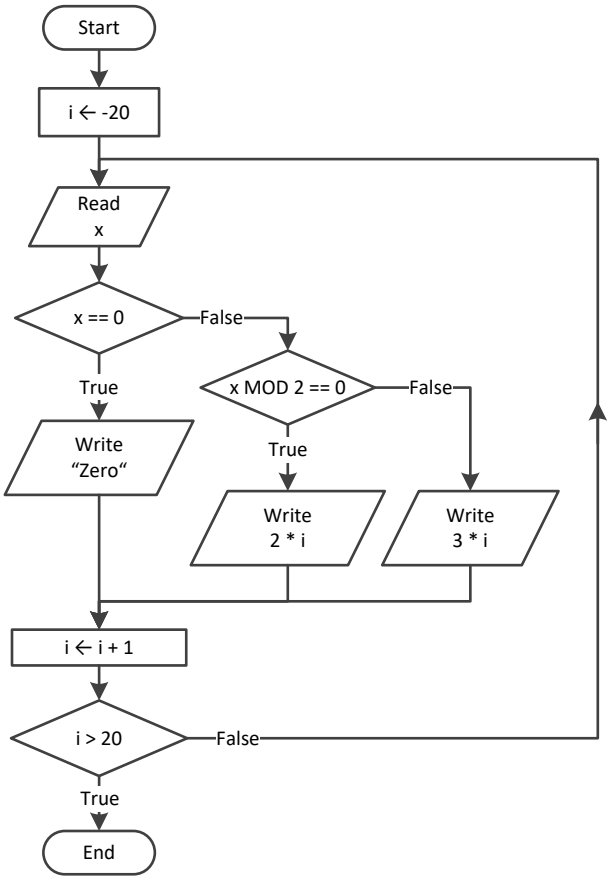
Chapter 27

27.4 Review Exercises

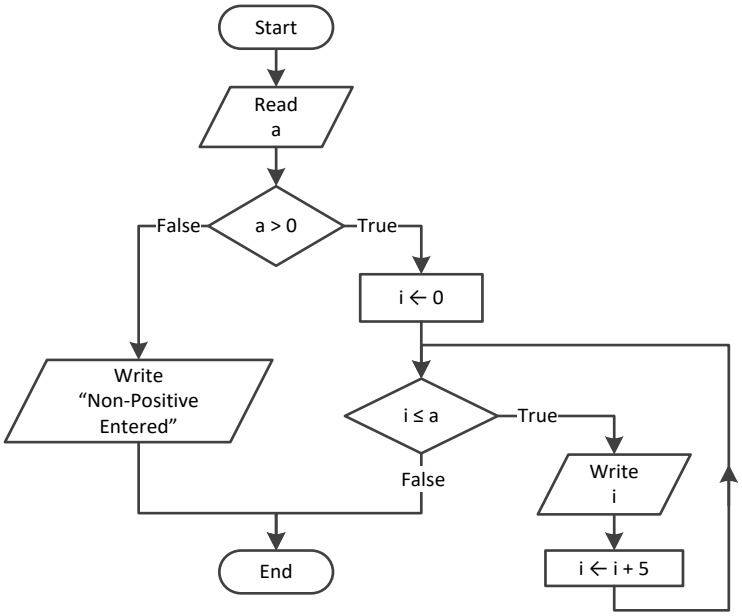
1. Solution



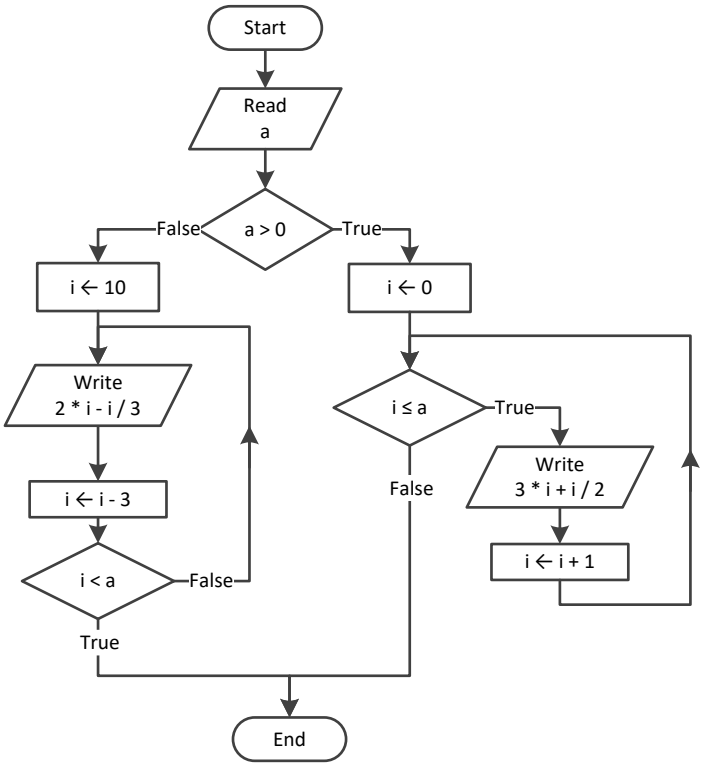
2. Solution



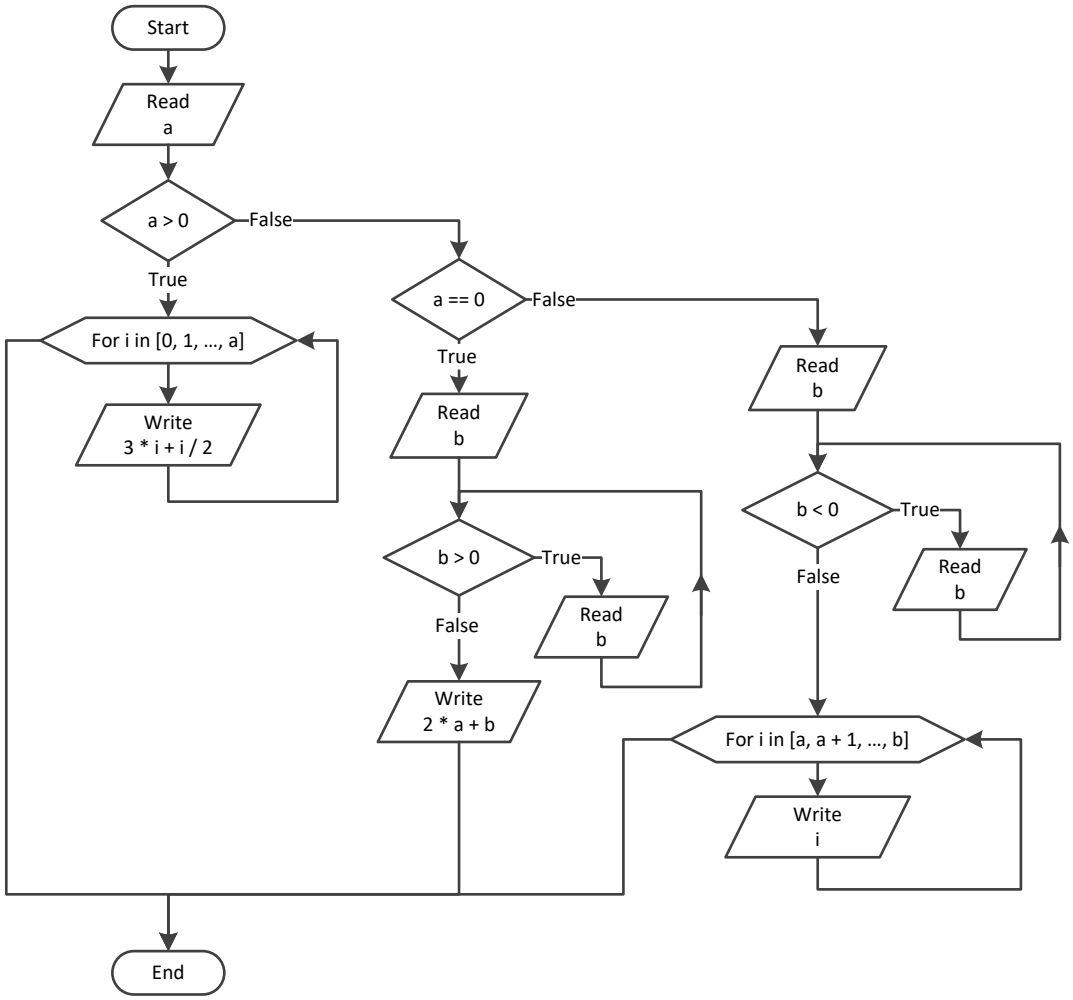
3. Solution



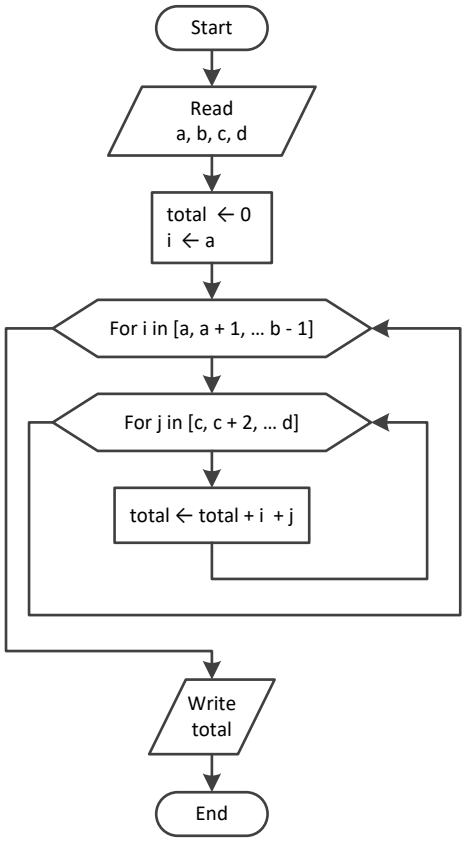
4. Solution


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

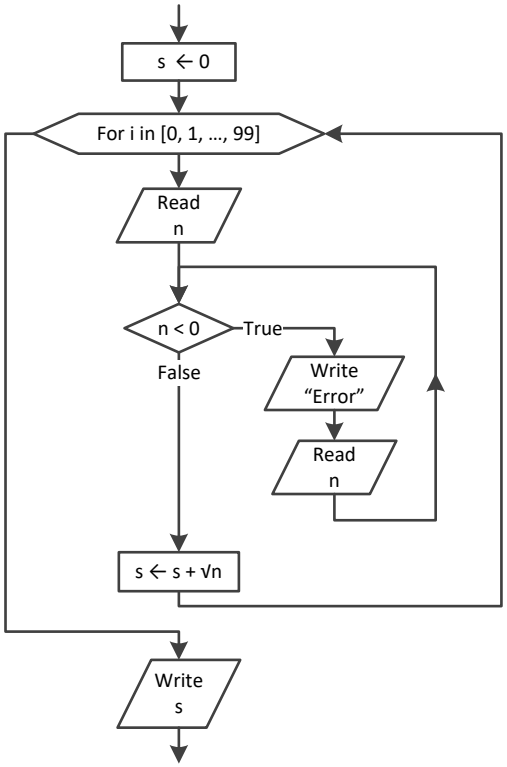
5. Solution


Start
Read
a
a > 0
False
True
For i in [0, 1, ..., a]
Write
3 * i + i / 2
a == 0
False
True
Read
b
b > 0
True
Read
b
False
Write
2 * a + b
Read
b
b < 0
True
Read
b
False
For i in [a, a + 1, ..., b]
Write
i
End

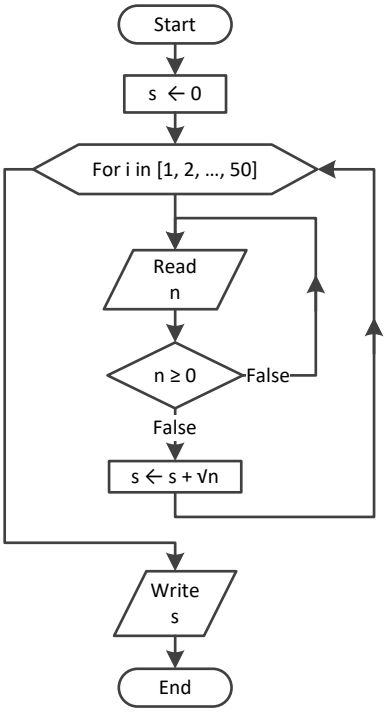
6. Solution


Start
Read
a, b, c, d
total ← 0
i ← a
For i in [a, a + 1, … b - 1]
For j in [c, c + 2, … d]
total ← total + i + j
Write
total
End

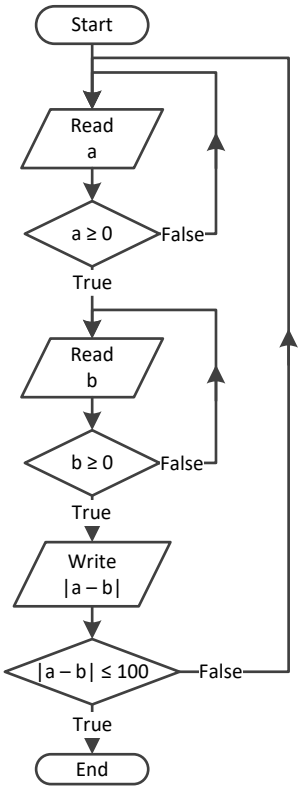
7. Solution



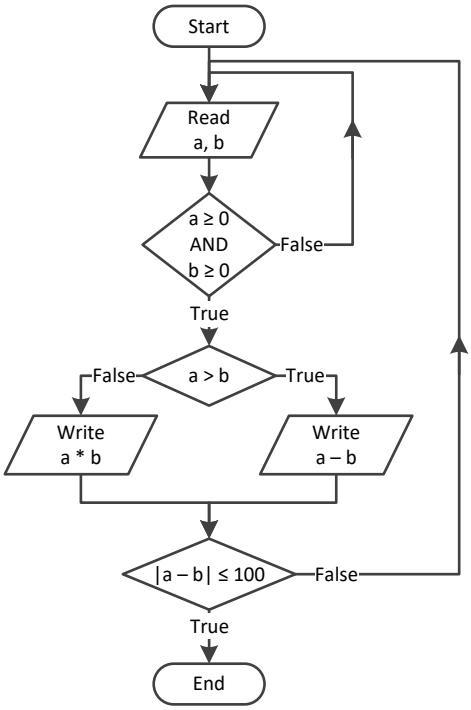
8. Solution



9. Solution



10. Solution



11. Solution

```
i = 0
a = int(input())
while True:
    if i % 2 != 0:
        print(i)
    i += 5
    if i >= a: break
```

12. Solution

```
a = int(input())
while a != -1:
    while True:
        b = int(input())
        if b > a: break
    i = a
    while i <= b:
        print(i)
        i += 1
    a = int(input())
```

13. Solution

```
i = 1
S = 0
P = 1
a = 0
while True:
    if i < 45:
        S += a
    else:
        P *= a
    i += 1
    if i >= 90: break
    a = float(input())

print(S, P)
```

Chapter 28

28.8 Review Questions: True/False

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

28.9 Review Questions: Multiple Choice

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

28.10 Review Exercises

1. Solution

```
countNames = 0
countNotJohns = 0

name = ""
name = input("Enter a name: ")
while name != "STOP":
    name = input("Enter a name: ")
    countNames += 1
    if name != "John":
        countNotJohns += 1
    name = input("Enter a name: ")

print("Total names entered:", countNames)
print("Names other than John entered:", countNotJohns)
```

2. Solution

First approach

```
text = input("Enter a text: ")

found = False
for character in text:
    if character == " ":
        found = True
        break

if found == False:
    print("One Single Word")
```

```
else:
    print("Complete Sentence")
```

Second approach

```
text = input("Enter a text: ")

if text.find(" ") == -1:
    print("One Single Word")
else:
    print("Complete Sentence")
```

Third approach

```
text = input("Enter a text: ")

if " " not in text:
    print("One Single Word")
else:
    print("Complete Sentence")
```

3. Solution

First approach

```
sentence = input("Enter a sentence: ")

found = False
for character in sentence:
    if character in "0123456789":
        found = True
        break

if found:
    print("The sentence contains a number")
```

Second approach

```
sentence = input("Enter a sentence: ")

found = False
for digit in "0123456789":
    if digit in sentence:
        found = True
        break

if found:
    print("The sentence contains a number")
```

4. Solution

```
print("Printing all integers from 1 to 100")
i = 1
while i < 101:
    print(i)
    i += 1
```

5. Solution

```
print("Printing odd integers from 1 to 99")
i = 1
while not(i > 100):
    print(i)
    i += 2
```

6. Solution

```
s = 0
i = 1
count = 100
while True:
    number = float(input())
    s = s + number
    i += 1
    if i > count: break
average = s / count
print(average)
```

7. Solution

```
s = 0

denom = 1
for j in range(1, 101):
    denom *= j

for i in range(1, 101):
    s += i / denom
print(s)
```

8. Solution

```
for i in range(1, 5):
    for j in range(1, 5):
        print(i, "x", j, "=", i * j)
```

9. Solution

```
print("\t|\t", end = "")
for i in range(1, 13):
    print(i, "\t", end = "")
print()

for i in range(1, 13):
    print("---------", end = "")
print()

for i in range(1, 13):
```

```
    print(i, "\t|\t", end = "")
    for j in range(1, 13):
        print(i * j, end = "\t")
    print()
```

10. Solution

```
n = int(input("Enter an integer: "))

print("\t|\t", end = "")
for i in range(1, n + 1):
    print(i, "\t", end = "")
print()

for i in range(n + 1):
    print("--------", end = "")
print()

for i in range(1, n + 1):
    print(i, "\t|\t", end = "")
    for j in range(1, n + 1):
        print(i * j, end = "\t")
    print()
```

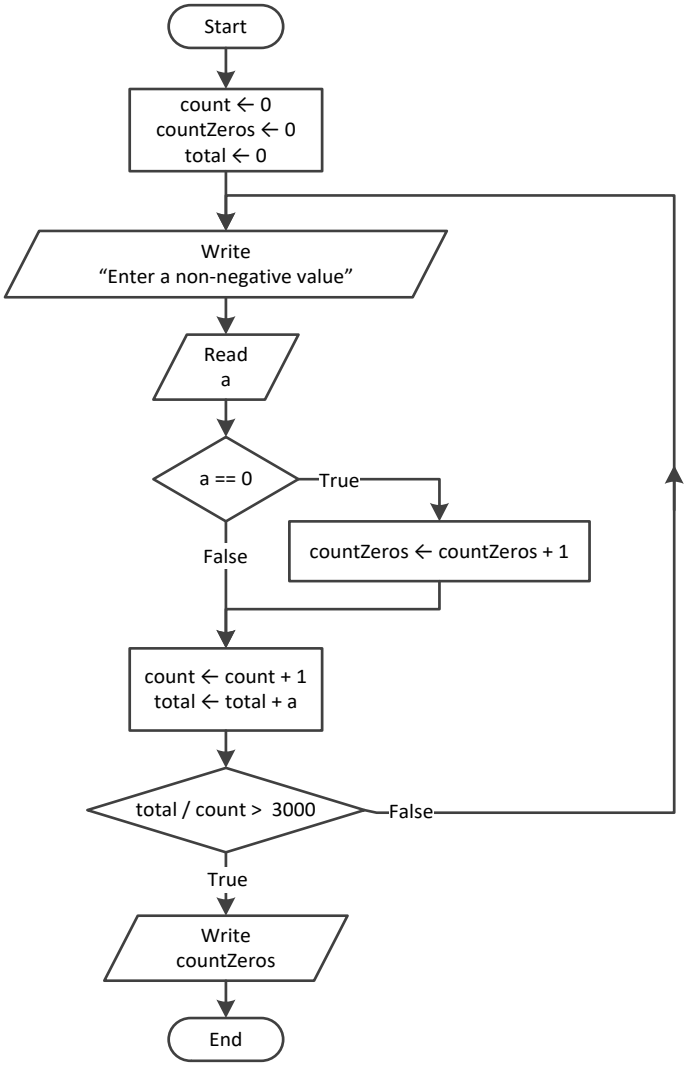
Chapter 29

29.7 Review Questions: True/False

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

29.8 Review Exercises

1. Solution




```
count = 0
countZeros = 0
total = 0

while True:
    a = float(input("Enter a non-negative value: "))
```

```
    if a == 0:
        countZeros += 1
    count += 1
    total += a
    if total / count > 3000: break

print(countZeros)
```

2. Solution

First approach

```
a = int(input("Enter an integer between 1 and 20: "))
for i in range(1000, 10000):
    d4 = i % 10
    r = i // 10
    d3 = r % 10
    r = r // 10
    d2 = r % 10
    d1 = r // 10
    if d1 + d2 + d3 + d4 < a:
        print(i)
```

However, using the `divmod()` function, it can become:

```
a = int(input("Enter an integer between 1 and 20: "))
for i in range(1000, 10000):
    r, d4 = divmod(i, 10)
    r, d3 = divmod(r, 10)
    d1, d2 = divmod(r, 10)

    if d1 + d2 + d3 + d4 < a:
        print(i)
```

Second approach

```
a = int(input("Enter an integer between 1 and 20: "))
for d1 in range(1, 10):
    for d2 in range(10):
        for d3 in range(10):
            for d4 in range(10):
                if d1 + d2 + d3 + d4 < a:
                    print(d1 * 1000 + d2 * 100 + d3 * 10 + d4)
```

Third approach

```
a = int(input("Enter an integer between 1 and 20: "))
for i in range(1000, 10000):
    d1, d2, d3, d4 = str(i)                         #d1, d2, d3, d4 are strings
    if int(d1) + int(d2) + int(d3) + int(d4) < a:  #In order to sum the digits, convert to integer
        print(i)
```

3. Solution

First approach

```
for i in range(1000, 10000):
```

```
    d1 = i // 1000
    r = i % 1000
    d2 = r // 100
    r = r % 100
    d3 = r // 10
    d4 = r % 10

    if d1 > d2 and d2 == d3 and d3 < d4:
        print(i)
```

However, using the `divmod()` function, it can become:

```
for i in range(1000, 10000):
    d1, r = divmod(i, 1000)
    d2, r = divmod(r, 100)
    d3, d4 = divmod(r, 10)
    if d1 > d2 and d2 == d3 and d3 < d4:
        print(i)
```

Second approach

```
for d1 in range(1, 10):
    for d2 in range(10):
        for d3 in range(10):
            for d4 in range(10):
                if d1 > d2 and d2 == d3 and d3 < d4:
                    print(d1 * 1000 + d2 * 100 + d3 * 10 + d4)
```

Third approach

```
for i in range(1000, 10000):
    d1, d2, d3, d4 = str(i)                    #d1, d2, d3, d4 are strings
    if d1 > d2 and d2 == d3 and d3 < d4:       #No need to convert to integer. Compare them as strings
        print(i)
```

4. Solution

First approach

```
x = int(input("Enter an integer: "))
count = 0

while x != 0:
    count += 1
    x = x // 10

print(count)
```

Second approach

```
x = int(input("Enter an integer: "))
#Convert the absolute value of x to string and get its length
count = len(str(abs(x)))
print(count)
```

5. Solution

```
x = int(input())
while x != 1 and x != 0:
    print("Error")
    x = int(input())
```

6. Solution

```
while True:
    gender = input().upper()
    if gender == "M" or gender == "F" or gender == "O": break
```

7. Solution

```
from math import sqrt

x = float(input("Enter a non-negative number: "))
count = 0
while x < 0:
    count += 1
    if count == 2: break

    print("Error: Invalid number!")
    x = float(input("Enter a non-negative number: "))

if count < 2:
    y = sqrt(x)
    print(y)
else:
    print("Dude, you are dumb!")
```

8. Solution

```
from math import pi

while True:
    r = float(input("Enter the length of a radius: "))
    while r <= 0:
        r = float(input("Invalid radius. Enter the length of a radius: "))

    area = pi * r ** 2
    print("The area is:", area)

    answer = input("Would you like to repeat? ")
    if answer.upper() != "YES": break
```

9. Solution

```
maximum = -460
```

```
total = 0
for i in range(31):
    t = float(input("Enter temperature for day " + str(i + 1) + ": "))
    while t < -459.67:
        print("Error! Wrong temperature.")
        t = float(input("Enter temperature for day " + str(i + 1) + ": "))

    total += t
    if t > maximum:
        maximum = t

print(total / 31, maximum)
```

10. Solution

```
level = float(input())
if level != 9999:
    hour = int(input())
    minutes = int(input())

    maximum = level
    maxHour = hour
    maxMinutes = minutes

    minimum = level
    minHour = hour
    minMinutes = minutes

    level = float(input())
    while level != 9999:
        hour = int(input())
        minutes = int(input())

        if level > maximum:
            maximum = level
            maxHour = hour
            maxMinutes = minutes

        if level < minimum:
            minimum = level
            minHour = hour
            minMinutes = minutes

        level = float(input())

    print(maximum, maxHour, maxMinutes, minimum, minHour, minMinutes)
```

11. Solution

```
number = int(input("Enter an integer: "))
```

```
while True:
    total = 0
    while number > 0:
        total += number % 10
        number //= 10

    if total > 26:
        number = total
    else:
        break

alphabet = "ABCDEFGHIJKLMNOPQRSTUVWXYZ"
print("The name of the person who might be thinking of you starts with a(an):", alphabet[total - 1])
```

12. Solution

```
for x in range(-100, 101):
    for y in range(-100, 101):
        if 5 * x + 3 * y ** 2 == 0:
            print(x, ",", y)
```

13. Solution

```
for x in range(-10, 11):
    for y in range(-10, 11):
        for z in range(-10, 11):
            if (x + y) / 2 + 3 * z ** 2 / (x + 3 * y + 45) == x / 3:
                print(x, ",", y, ",", z)
```

14. Solution

```
m1 = int(input())
m2 = int(input())
m3 = int(input())

s = 0
while m2 != 0:
    if m2 % 2 != 0:
        s += m1
    m1 *= 2
    m2 = m2 // 2

m1 = s
m2 = m3

s = 0
while m2 != 0:
    if m2 % 2 != 0:
        s += m1
    m1 *= 2
```

```
    m2 = m2 // 2

print(s)
```

15. Solution

```
a = float(input())
while a <= 0 or a != int(a):
    print("Error! You must enter a positive integer")
    a = float(input())
x = int(a)

numberOfDivisors = 2
for i in range(2, x // 2 + 1):
    if x % i == 0:
        numberOfDivisors += 1

print(numberOfDivisors)
```

16. Solution

```
x = int(input("Enter an integer greater than 1: "))
while x <= 1:
    print("Error!")
    x = int(input("Enter an integer greater than 1: "))

numberOfDivisors = 2
for i in range(2, x // 2 + 1):
    if x % i == 0:
        numberOfDivisors += 1
        break

if numberOfDivisors == 2:
    print("Number", x, "is prime")
```

17. Solution

```
from math import sqrt

start = int(input("Enter an positive integer: "))
finish = int(input("Enter a second positive integer: "))
if start > finish:
    c = start           # Or you can do the following:
    start = finish      # start, finish = finish, start
    finish = c

for x in range(start, finish + 1):
    for y in range(x, finish + 1):
        z = sqrt(x ** 2 + y ** 2)
        #If result is integer and less than or equal to finish, display x, y, z
```

```
        if z == int(z) and z <= finish:
            print(x, y, int(z))
```

18. Solution

```
a = int(input("Enter an integer greater than 1: "))
while a < 2:
    a = int(input("Wrong number. Please enter an integer greater than 1: "))

b = int(input("Enter a second integer greater than 1: "))
while b < 2:
    b = int(input("Wrong number. Please enter a second integer greater than 1: "))

if a > b:
    c = a           # Or you can do the following:
    a = b           # a, b = b, a
    b = c

for x in range(a, b + 1):
    numberOfDivisors = 2
    i = 2
    while i <= x // 2 and numberOfDivisors == 2:
        if x % i == 0:
            numberOfDivisors += 1
        i += 1
    if numberOfDivisors == 2:
        print("Number", x, "is prime")
```

19. Solution

```
y = float(input("Enter a positive integer: "))
while y <= 0 or y != int(y):
    y = float(input("Wrong number! Enter a positive integer: "))
x = int(y)

total = 0
for i in range(1, x):
    if x % i == 0:
        total += i

if total == x:
    print("Number", x, "is a perfect number")
else:
    print("Number", x, "is not a perfect number")
```

20. Solution

```
y = float(input("Enter a positive integer: "))
while y <= 0 or y != float(y):
    y = float(input("Wrong number! Enter a positive integer: "))
```

```
a = int(y)

y = float(input("Enter a second positive integer: "))
while y <= 0 or y != float(y):
    y = float(input("Wrong number! Enter a second positive integer: "))
b = int(y)

if a > b:
    c = a           # Or you can do the following:
    a = b           # a, b = b, a
    b = c

for x in range(a, b + 1):
    total = 0
    for j in range(1, x):
        if x % j == 0:
            total += j

    if total == x:
        print("Number", x, "is a perfect number")
```

21. Solution

First approach

```
a = int(input("Enter a positive four-digit integer: "))
while a < 1000 or a > 9999:
    a = int(input("Wrong number. Please enter a positive four-digit integer: "))

b = int(input("Enter a second positive four-digit integer: "))
while b < 1000 or b > 9999:
    b = int(input("Wrong number. Please enter a second positive four-digit integer: "))

if a > b:
    c = a           # Or you can do the following:
    a = b           # a, b = b, a
    b = c

for x in range(a, b + 1):
    r, d4 = divmod(x, 10)
    r, d3 = divmod(r, 10)
    d1, d2 = divmod(r, 10)

    if d1 == d4 and d2 == d3:
        print(x)
```

Second approach

```
a = int(input("Enter a positive four-digit integer: "))
while a < 1000 or a > 9999:
    a = int(input("Wrong number. Please enter a positive four-digit integer: "))

b = int(input("Enter a second positive four-digit integer: "))
```

```
while b < 1000 or b > 9999:
    b = int(input("Wrong number. Please enter a second positive four-digit integer: "))

if a > b:
    a, b = b, a

for x in range(a, b + 1):
    if str(x) == str(x)[::-1]:
        print(x)
```

22. Solution

```
for i in range(31):
    print(2 ** i)
```

23. Solution

```
offset = 10
i = 1
while i <= 401:
    print(i)
    i += offset
    offset += 2
```

24. Solution

```
for i in range(1, 101):
    print(-i, "\n", i)
```

25. Solution

First approach

```
value = 0
for i in range(8):
    offset = 10 ** i
    value += offset
    print(value)
```

Second approach

```
value = "1"
for i in range(8):
    print(value)
    value += "1"
```

26. Solution

First approach

```
a = int(input())

fibPrevious2 = 0
fibPrevious = 1
```

```
fib = 1
for i in range(a):
    print(fibPrevious2)
    fib = fibPrevious + fibPrevious2
    fibPrevious2 = fibPrevious
    fibPrevious = fib
```

Second approach

```
a = int(input())

fibPrevious2, fibPrevious, fib = 0, 1, 1
for i in range(a):
    print(fibPrevious2)
    fib = fibPrevious + fibPrevious2
    fibPrevious2, fibPrevious = fibPrevious, fib
```

27. Solution

First approach

```
a = int(input())

fibPrevPrev = 0
fibPrev = 1
fib = 1
while fib < a:
    print(fib)
    fib = fibPrev + fibPrevPrev
    fibPrevPrev = fibPrev
    fibPrev = fib
```

Second approach

```
a = int(input())

fibPrevPrev, fibPrev, fib = 0, 1, 1
while fib < a:
    print(fib)
    fib = fibPrev + fibPrevPrev
    fibPrevPrev, fibPrev = fibPrev, fib
```

28. Solution

```
n = int(input("Enter a positive integer: "))
while n <= 0:
    n = int(input("Wrong number. Please enter a positive integer: "))

nominator = 0
for i in range(2, 2 * n + 2, 2):
    nominator += i

denominator = 1
for i in range(1, n + 1):
```

```
    denominator *= i

y = nominator / denominator
print(y)
```

29. Solution

```
n = int(input("Enter a positive integer: "))
while n <= 0:
    n = int(input("Wrong number. Please enter a positive integer: "))

nominator = 0
sign = 1
for i in range(1, 2 * n + 1 + 2, 2):
    nominator += sign * i
    sign = -sign

y = nominator / n
print(y)
```

30. Solution

```
n = int(input("Enter an integer greater than 2: "))
while n <= 2:
    n = int(input("Wrong number. Please enter an integer greater than 2: "))

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

sign = 1
for i in range(3, n + 2, 2):
    y += sign / i
    sign = -sign

print(y)
```

31. Solution

```
n = int(input("Enter a positive integer: "))
while n <= 0:
    n = int(input("Wrong number. Please enter a positive integer: "))

y = 0
for i in range(1, n + 1):
    y += 1 / i ** (n - i + 1)

print(y)
```

32. Solution

```
n = int(input("Enter a non-negative integer: "))
```

```
factorial = 1
for i in range(1, n + 1):
    factorial *= i

print(factorial)
```

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

33. Solution

First approach

```
ACCURACY = 0.00001

x = float(input())

exponential = 0
i = 0
while True:
    exponentialPrevious = exponential

    factorial = 1
    for j in range(1, i + 1):
        factorial *= j

    exponential += x ** i / factorial

    i += 1
    if abs(exponential - exponentialPrevious) <= ACCURACY: break

print("e(", x, ") ~=", exponential)
```

Second approach

```
ACCURACY = 0.00001

x = float(input())

exponential = 1
i = 1
factorial = 1
while True:
    exponentialPrevious = exponential

    factorial *= i

    exponential += x ** i / factorial

    i += 1
    if abs(exponential - exponentialPrevious) <= ACCURACY: break
```

```
print("e(", x, ") ~=", exponential)
```

34. Solution

First approach

```
ACCURACY = 0.00001

x = float(input())

sign = 1
sinus = 0
i = 1
while True:
    sinusPrevious = sinus

    factorial = 1
    for j in range(1, i + 1):
        factorial *= j

    sinus += sign * x ** i / factorial

    sign = -sign
    i += 2
    if abs(sinus - sinusPrevious) <= ACCURACY: break

print("sin(", x, ") ~=", sinus)
```

Second approach

```
ACCURACY = 0.00001

x = float(input())

sign = -1
sinus = x
i = 3
factorial = 1
while True:
    sinusPrevious = sinus

    factorial *= i * (i - 1)

    sinus += sign * x ** i / factorial

    sign = -sign
    i += 2
    if abs(sinus - sinusPrevious) <= ACCURACY: break

print("sin(", x, ") ~=", sinus)
```

35. Solution

First approach

```
ACCURACY = 0.00001

x = float(input())

sign = 1
cosinus = 0
i = 0
while True:
    cosinusPrevious = cosinus

    factorial = 1
    for j in range(1, i + 1):
        factorial *= j

    cosinus += sign * x ** i / factorial

    sign = -sign
    i += 2
    if abs(cosinus - cosinusPrevious) <= ACCURACY: break

print("cos(", x, ") ~=", cosinus)
```

Second approach

```
ACCURACY = 0.00001

x = float(input())

sign = -1
cosinus = 1
i = 2
factorial = 1
while True:
    cosinusPrevious = cosinus

    factorial *= i * (i - 1)

    cosinus += sign * x ** i / factorial

    sign = -sign
    i += 2
    if abs(cosinus - cosinusPrevious) <= ACCURACY: break

print("cos(", x, ") ~=", cosinus)
```

36. Solution

```
alphabet = "abcdefghijklmnopqrstuvwxyz"
```

```
while True:
    a = int(input("Enter an integer between 1 and 26: "))

    failure = False
    if a < 1:
        print("Please enter positive integers!")
        failure = True
    elif a > 26:
        print("Please enter a value less than or equal to 26!")
        failure = True
    if not failure: break

while True:
    b = int(input("Enter an integer between 1 and 26: "))

    failure = False
    if b < 1:
        print("Please enter positive integers!")
        failure = True
    elif b > 26:
        print("Please enter a value less than or equal to 26!")
        failure = True
    if not failure: break

if a > b:
    a, b = b, a

for i in range(a, b + 1):
    print(alphabet[i - 1])
```

37. Solution

```
from random import randrange

secretNumber = randrange(1, 101)

attempts = 1
guess = int(input("Enter a guess: "))
while guess != secretNumber:
    if guess > secretNumber:
        print("Your guess is bigger than my secret number. Try again.")
    else:
        print("Your guess is smaller than my secret number. Try again.")
    attempts += 1
    guess = int(input("Enter a guess: "))

print("You found it!")
print("Attempts:", attempts)
```

38. Solution

```
from random import randrange

for i in range(2):
    secretNumber = randrange(1, 101)

    attempts = 1
    guess = int(input("Enter a guess: "))
    while guess != secretNumber:
        if guess > secretNumber:
            print("Your guess is bigger than my secret number. Try again.")
        else:
            print("Your guess is smaller than my secret number. Try again.")
        attempts += 1
        guess = int(input("Enter a guess: "))
    print("You found it!")
    print("Attempts:", attempts)

    if i == 0:
        firstPlayerAttempts = attempts

if firstPlayerAttempts < attempts:
    print("First player wins!")
elif firstPlayerAttempts > attempts:
    print("Second player wins!")
else:
    print("It's a draw")
```

39. Solution

```
while True:
    print("1. 4/3 TV Screen")
    print("2. 16/9 TV Screen")
    print("3. Exit")
    choice = int(input("Enter a choice: "))


    if choice == 1:
        diagonal = int(input("Enter diagonal: "))
        print("Width:", diagonal * 0.8)
        print("Height:", diagonal * 0.6)
    elif choice == 2:
        diagonal = int(input("Enter diagonal: "))
        print("Width:", diagonal * 0.87)
        print("Height:", diagonal * 0.49)
    elif choice == 3:
        break
```

40. Solution

```
n = int(input("Enter total number of students: "))
while n <= 0:
    n = int(input("Wrong number. Please enter total number of students: "))

total = 0
totalA = 0
countA = 0
totalB = 0
countB = 0
totalABoys = 0
countABoys = 0
countCdefGirls = 0

maximum = -1
minimum = 101

for i in range(n):
    grade = int(input("Enter grade for student No" + str(i + 1) + ": "))
    while grade < 0 or grade > 100:
        grade = int(input("Wrong grade. Please enter grade for student No" + str(i + 1) + ": "))

    gender = input("Enter gender for student No" + str(i + 1) + ": ").upper()
    while gender != "M" and gender != "F" and gender != "O":
        gender = input("Wrong gender. Please enter gender for student No" + str(i + 1) + ": ").upper()

    if 90 <= grade <= 100:
        totalA += grade
        countA += 1
        if gender == "M":
            totalABoys += grade
            countABoys += 1
    elif 80 <= grade <= 89:
        totalB += grade
        countB += 1
    else:
        if gender == "F":
            countCdefGirls += 1

    if grade > maximum:
        maximum = grade

    if grade < minimum:
        minimum = grade

    total += grade

if countA > 0:
```

```
    print("The average value of those who got an 'A' is: ")
    print(totalA / countA)
if countB > 0:
    print("The average value of those who got a 'B' is: ")
    print(totalB / countB)
if countABoys > 0:
    print("The average value of boys who got an 'A' is: ")
    print(totalABoys / countABoys)

print("The total number of girls that got less than 'B' is:", countCdefGirls)
print("The highest grade is:", maximum)
print("The lowest grade is:", minimum)
print("The average grade of the whole class is:", total / n)
```

41. Solution

```
while True:
    amount = float(input("Enter amount: "))
    while amount <= 0:
        amount = float(input("Wrong amount. Please enter amount: "))

    if amount < 20:
        discount = 0
    elif amount < 50:
        discount = 3
    elif amount < 100:
        discount = 5
    else:
        discount = 10

    print("Discount:", discount, "%", sep = "")
    print("Amount to pay (discount included):", (amount - amount * discount / 100))

    answer = input("Would you like to repeat? ")
    if answer.upper() != "YES": break
```

42. Solution

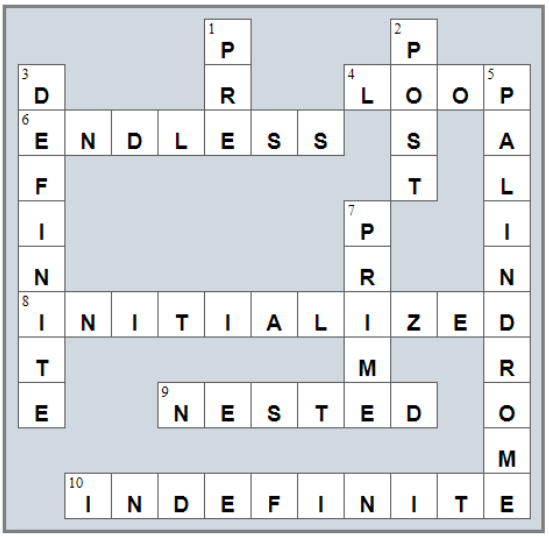
```
TAX_RATE = 0.25

kwh = int(input("Enter number of Kilowatt-hours consumed: "))
while kwh < 0 and kwh != -1:
    kwh = int(input("Wrong value. Please enter number of Kilowatt-hours consumed: "))

while kwh != -1:
    if kwh <= 400:
        t = kwh * 0.11
    elif kwh <= 1500:
        t = 400 * 0.11 + (kwh - 400) * 0.22
    elif 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
print("Total amount to pay (taxes included):", t)

kwh = int(input("Enter number of Kilowatt-hours consumed: "))
while kwh < 0 and kwh != -1:
    kwh = int(input("Wrong value. Please enter number of Kilowatt-hours consumed: "))
```

Review in “Loop Control Structures”

Review Crossword Puzzle

1.



Chapter 30

30.15 Review Questions: True/False

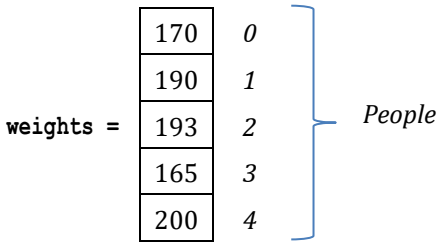
1. true
2. true
3. false
4. false
5. false
6. true
7. false
8. true
9. false
10. true
11. true
12. true
13. false
14. false
15. false
16. true
17. false
18. true
19. true
20. false
21. true
22. false
23. true
24. false
25. true
26. false
27. false
28. true
29. false
30. true
31. true
32. false
33. false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

30.16 Review Questions: Multiple Choice

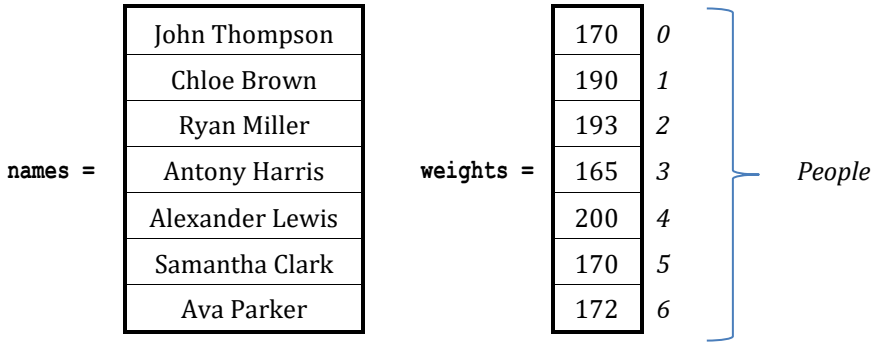
1. b
2. a
3. c
4. b
5. d
6. b
7. d
8. d
9. c
10. a
11. b
12. a
13. b
14. b

30.17 Review Exercises

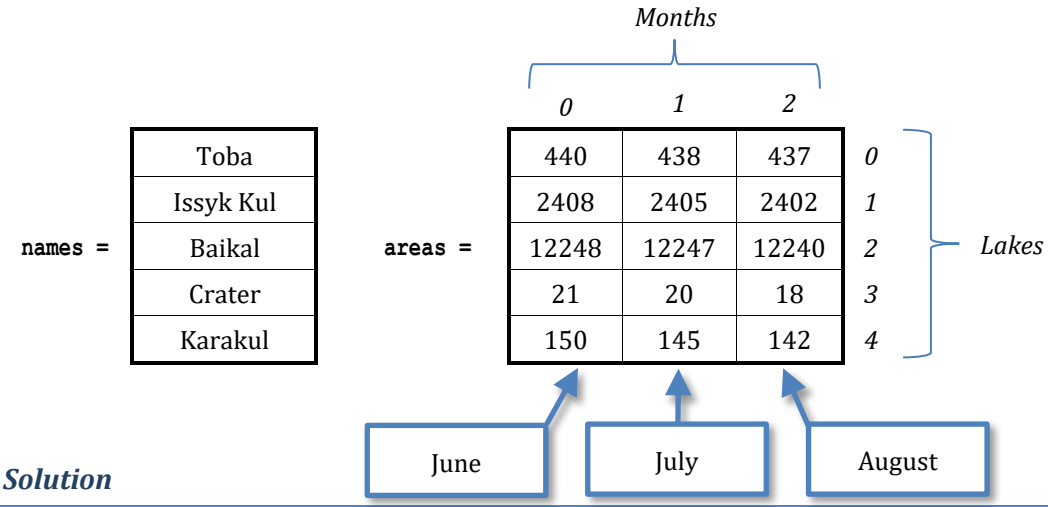
1. Solution



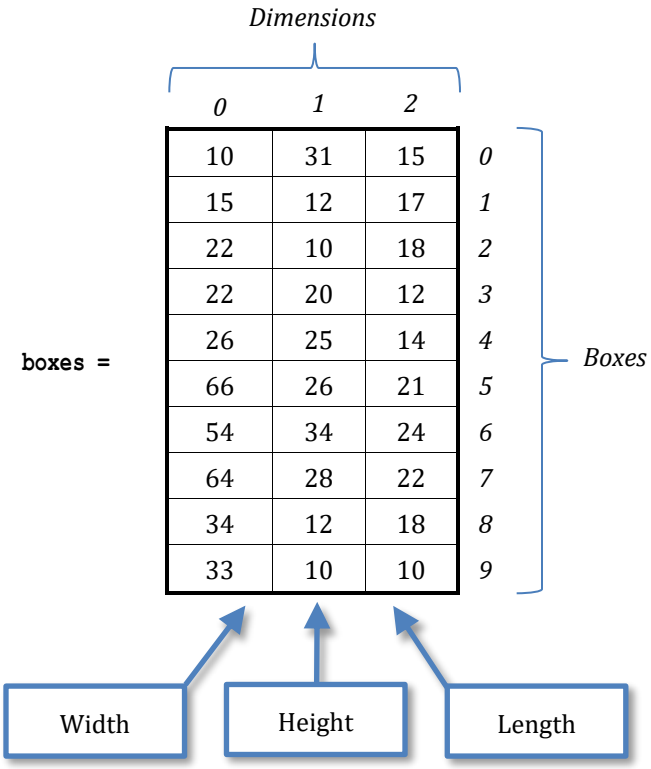
2. Solution



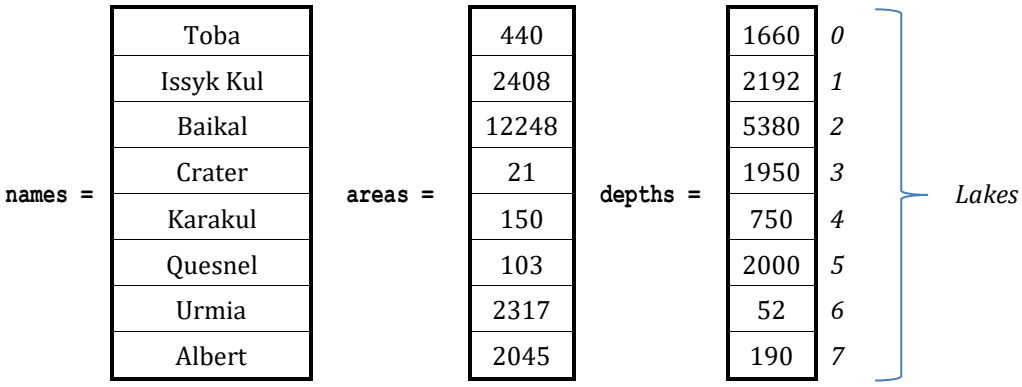
3. Solution



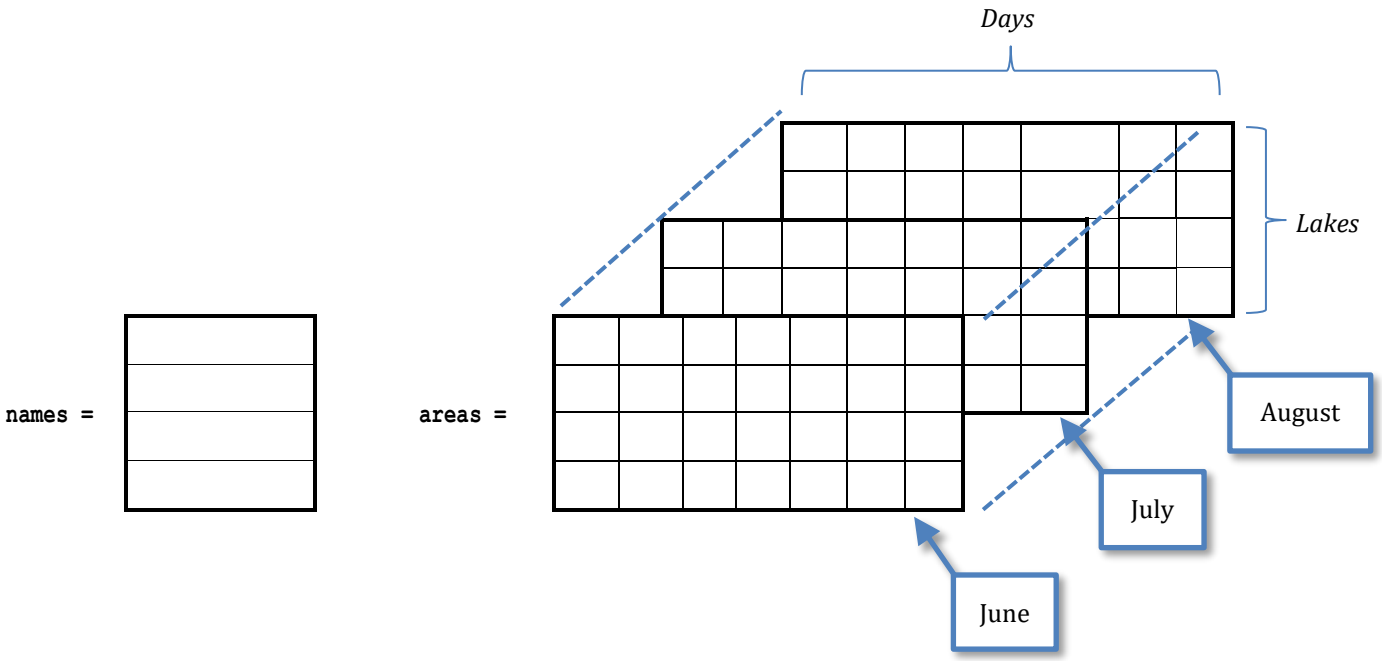
4. Solution



5. Solution



6. Solution



7. Solution

Step	Statement	x	a[0]	a[1]	a[2]
1	`a = [None] * 3`	?	?	?	?
2	`a[2] = 1`	?	?	?	**1**
3	`x = 0`	**0**	?	?	1
4	`a[x + a[2]] = 4`	0	?	**4**	1
5	`a[x] = a[x + 1] * 4`	0	**16**	4	1

8. Solution

Step	Statement	x	a[0]	a[1]	a[2]	a[3]	a[4]
1	a = [None] * 5	?	?	?	?	?	?
2	a[1] = 5	?	?	**5**	?	?	?
3	x = 0	**0**	?	5	?	?	?
4	a[x] = 4	0	**4**	5	?	?	?
5	a[a[0]] = a[x + 1] % 3	0	4	5	?	?	**2**
6	a[a[0] / 2] = 10	0	4	5	**10**	?	2
7	x += 2	**2**	4	5	10	?	2
8	a[x + 1] = a[x] + 9	2	4	5	10	**19**	2

9. Solution

For input value of 3

Step	Statement	x	a[0]	a[1]	a[2]	a[3]
1	a = [None] * 4	?	?	?	?	?
2	a[1] = int(input())	?	?	**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	**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 = [None] * 4	?	?	?	?	?
2	a[1] = int(input())	?	?	**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	**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 = [None] * 4	?	?	?	?	?
2	a[1] = int(input())	?	?	**1**	?	?
3	x = 0	**0**	?	1	?	?

4	`a[x] = 3`	0	**3**	1	?	?
5	`a[a[0]] = a[x + 1] % 2`	0	3	1	?	**3**
6	`a[a[0] % 2] = 10`	0	3	**10**	?	3
7	`x += 1`	**1**	3	10	?	3
8	`a[x + 1] = a[x] + 9`	1	3	10	**19**	3

10. Solution

For input value of 100

Step	Statement	x	a[0]	a[1]	a[2]	a[3]
1	`a = [None] * 4`	?	?	?	?	?
2	`a[1] = int(input())`	?	?	**100**	?	?
3	`x = 0`	**0**	?	100	?	?
4	`a[x] = 3`	0	**3**	100	?	?
5	`a[a[0]] = a[x + 1] % 10`	0	3	100	?	**0**
6	`if a[3] > 5:`			False		
7	`a[2] = 3`	0	3	100	**3**	0

For input value of 108

Step	Statement	x	a[0]	a[1]	a[2]	a[3]
1	`a = [None] * 4`	?	?	?	?	?
2	`a[1] = int(input())`	?	?	**108**	?	?
3	`x = 0`	**0**	?	108	?	?
4	`a[x] = 3`	0	**3**	108	?	?
5	`a[a[0]] = a[x + 1] % 10`	0	3	108	?	**8**
6	`if a[3] > 5:`			True		
7	`a[a[0] % 2] = 9`	0	3	**9**	?	8
8	`x += 1`	**1**	3	9	?	8
9	`a[x + 1] = a[x] + 9`	1	3	9	**18**	8

For input value of 1

Step	Statement	x	a[0]	a[1]	a[2]	a[3]
1	`a = [None] * 4`	?	?	?	?	?
2	`a[1] = int(input())`	?	?	**1**	?	?
3	`x = 0`	**0**	?	1	?	?
4	`a[x] = 3`	0	**3**	1	?	?
5	`a[a[0]] = a[x + 1] % 10`	0	3	1	?	**1**
6	`if a[3] > 5:`			False		
7	`a[2] = 3`	0	3	1	**3**	1

11. Solution

Step	Statement	x	y	a[0]	a[1]	a[2]
1	`a = [None] * 3`	?	?	?	?	?
2	`x = 4`	**4**	?	?	?	?
3	`y = x - 1`	4	**3**	?	?	?

4, 5	`if x > y:`
	`  a[0] = 1`
	`else:`
	`  a[0] = y`
		4	3	**1**	?	?

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

12. Solution

Step	Statement	i	a[0]	a[1]	a[2]	a[3]	a[4]	a[5]
1	`a = [17, 12, 45, 12, 12, 49]`	?	**17**	**12**	**45**	**12**	**12**	**49**
2	`i = 0`	**0**	17	12	45	12	12	49
3	`if a[i] == 12:`	False						
4	`a[i] += 1`	0	**18**	12	45	12	12	49
5	`i = 1`	**1**	18	12	45	12	12	49
6	`if a[i] == 12:`	True						
7	`a[i] -= 1`	1	18	**11**	45	12	12	49
8	`i = 2`	**2**	18	11	45	12	12	49
9	`if a[i] == 12:`	False						
10	`a[i] += 1`	2	18	11	**46**	12	12	49
11	`i = 3`	**3**	18	11	46	12	12	49
12	`if a[i] == 12:`	True						
13	`a[i] -= 1`	3	18	11	46	**11**	12	49
14	`i = 4`	**4**	18	11	46	11	12	49
15	`if a[i] == 12:`	True						
16	`a[i] -= 1`	4	18	11	46	11	**11**	49
17	`i = 5`	**5**	18	11	46	11	11	49
18	`if a[i] == 12:`	False						
19	`a[i] += 1`	5	18	11	46	11	11	**50**

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	`a[i] = a[i + 1] + a[i - 1]`	1	10	**22**	12	23	22	19
4	`i = 2`	**2**	10	22	12	23	22	19
5	`a[i] = a[i + 1] + a[i - 1]`	2	10	22	**45**	23	22	19
6	`i = 3`	**3**	10	22	45	23	22	19
7	`a[i] = a[i + 1] + a[i - 1]`	3	10	22	45	**67**	22	19
8	`i = 4`	**4**	10	22	45	67	22	19
9	`a[i] = a[i + 1] + a[i - 1]`	4	10	22	45	67	**86**	19

14. Solution

It displays:
Navajo
Cherokee
Sioux

15. Solution

First approach

```
ELEMENTS = 100

a = [None] * ELEMENTS
for i in range(ELEMENTS):
    a[i] = float(input())

for i in range(ELEMENTS):
    print(a[i] ** 3)
```

Second approach

```
ELEMENTS = 100

a = []
for i in range(ELEMENTS):
    a.append(float(input()))

for element in a:
    print(element ** 3)
```

16. Solution

First approach

```
ELEMENTS = 80

a = [None] * ELEMENTS
for i in range(ELEMENTS):
```

```
    a[i] = float(input())

for i in range(ELEMENTS):
    a[i] = a[i] ** 2

for i in range(ELEMENTS - 1, -1, -1):
    print(a[i])
```

Second approach

```
ELEMENTS = 80

a = []
for i in range(ELEMENTS):
    a.append(float(input()))

for i in range(ELEMENTS):
    a[i] = a[i] ** 2

for element in a[::-1]:
    print(element)
```

17. Solution

First approach

```
ELEMENTS = 90

a = [None] * ELEMENTS
for i in range(ELEMENTS):
    a[i] = int(input())

for i in range(ELEMENTS - 1, -1, -1):
    if a[i] % 5 == 0:
        print(a[i])
```

Second approach

```
ELEMENTS = 90

a = []
for i in range(ELEMENTS):
    a.append(int(input()))

for element in a[::-1]:
    if element % 5 == 0:
        print(element)
```

18. Solution

First approach

```
ELEMENTS = 50

a = [None] * ELEMENTS
for i in range(ELEMENTS):
    a[i] = int(input())
```

```
for i in range(ELEMENTS):
    if a[i] % 2 == 0 or a[i] > 10:
        print(a[i])
```

Second approach

```
ELEMENTS = 50

a = []
for i in range(ELEMENTS):
    a.append(int(input()))

for element in a:
    if element % 2 == 0 or element > 10:
        print(element)
```

19. Solution

First approach

```
ELEMENTS = 30

a = [None] * ELEMENTS
for i in range(ELEMENTS):
    a[i] = float(input())

total = 0
for i in range(ELEMENTS):
    if a[i] > 0:
        total += a[i]

print(total)
```

Second approach

```
ELEMENTS = 30

a = []
for i in range(ELEMENTS):
    a.append(float(input()))

total = 0
for element in a:
    if element > 0:
        total += element

print(total)
```

20. Solution

First approach

```
ELEMENTS = 50

a = [None] * ELEMENTS
for i in range(ELEMENTS):
```

```
    a[i] = int(input())

total = 0
for i in range(ELEMENTS):
    if  a[i] >= 10 and a[i] <= 99:
        total += a[i]

print(total)
```

Second approach

```
ELEMENTS = 50

a = []
for i in range(ELEMENTS):
    a.append(int(input()))

total = 0
for element in a:
    if 10 <= element <= 99:
        total += element

print(total)
```

21. Solution

First approach

```
ELEMENTS = 40

a = [None] * ELEMENTS
for i in range(ELEMENTS):
    a[i] = float(input())

sumPos = 0
sumNeg = 0
for i in range(ELEMENTS):
    if a[i] > 0 :
        sumPos += a[i]
    elif a[i] < 0:
        sumNeg += a[i]

print(sumPos, sumNeg)
```

Second approach

```
ELEMENTS = 40

a = []
for i in range(ELEMENTS):
    a.append(float(input()))

sumPos = sumNeg = 0
for element in a:
    if element > 0 :
        sumPos += element
```

```
    elif element < 0:
        sumNeg += element

print(sumPos, sumNeg)
```

22. Solution

First approach

```
ELEMENTS = 20

a = [None] * ELEMENTS
for i in range(ELEMENTS):
    a[i] = float(input())

total = 0
for i in range(ELEMENTS):
    total += a[i]

print(total / ELEMENTS)
```

Second approach

```
ELEMENTS = 20

a = []
for i in range(ELEMENTS):
    a.append(float(input()))

total = 0
for element in a:
    total += element

print(total / ELEMENTS)
```

23. Solution

```
ELEMENTS = 50

a = [None] * ELEMENTS
for i in range(ELEMENTS):
    a[i] = int(input("Enter an integer: "))

for i in range(ELEMENTS):
    if a[i] < 20:
        print(i)
```

24. Solution

```
ELEMENTS = 60

a = [None] * ELEMENTS
for i in range(ELEMENTS):
    a[i] = float(input("Enter a number: "))

for i in range(0, ELEMENTS, 2):
```

```
    print(a[i])
```

25. Solution

```
ELEMENTS = 20

a = [None] * ELEMENTS
for i in range(ELEMENTS):
    a[i] = float(input("Enter a number: "))

total = 0
for i in range(0, ELEMENTS, 2):
    total += a[i]

print(total)
```

26. Solution

First approach

```
ELEMENTS = 100

a = [None] * ELEMENTS
for i in range(ELEMENTS):
    a[i] = i + 1
```

Second approach

```
ELEMENTS = 100

a = []
for i in range(ELEMENTS):
    a.append(i + 1)
```

27. Solution

```
a = []
for i in range(2, 202, 2):
    a.append(i)
```

28. Solution

First approach

```
n = int(input("Enter N: "))
while n < 1:
    print("Error! Value must be greater than or equal to 1")
    n = int(input("Enter N: "))

a = [None] * n
for i in range(1, n + 1):
    a[i - 1] = i ** 2

for i in range(n):
    print(a[i])
```

Second approach

```
n = int(input("Enter N: "))
while n < 1:
    print("Error! Value must be greater than or equal to 1")
```

```
    n = int(input("Enter N: "))

a = []
for i in range(1, n + 1):
    a.append(i ** 2)

for element in a:
    print(element)
```

29. Solution

```
ELEMENTS = 10

a = []
for i in range(ELEMENTS):
    a.append(float(input("Enter a number: ")))

for i in range(ELEMENTS):
    if a[i] == int(a[i]):
        print(i)
```

30. Solution

```
ELEMENTS = 50

a = []
for i in range(ELEMENTS):
    a.append(float(input("Enter a number: ")))

count = 0
for i in range(ELEMENTS):
    if a[i] < 0:
        count += 1

print(count)
```

31. Solution

First approach

```
WORDS = 50

a = [None] * WORDS
for i in range(WORDS):
    a[i] = input()

for i in range(WORDS):
    if len(a[i]) >= 10:
        print(a[i])
```

Second approach

```
WORDS = 50

a = []
for i in range(WORDS):
```

```
    a.append(input())

for element in a:
    if len(element) >= 10:
        print(element)
```

32. Solution

```
ELEMENTS = 30

words = []
for i in range(ELEMENTS):
    words.append(input())

lengthLimits = (0, 5, 10, 20)

for i in range(1, 4):
    for word in words:
        if len(word) >= lengthLimits[i - 1] and len(word) < lengthLimits[i]:
            print(word)
```

33. Solution

First approach

```
WORDS = 40

a = [None] * WORDS
for i in range(WORDS):
    a[i] = input("Enter a word: ")

for i in range(WORDS):
    count = 0
    for j in range(len(a[i])):
        if a[i][j] == "w":
            count += 1
    if count >= 2:
        print(a[i])
```

Second approach

```
WORDS = 40

a = []
for i in range(WORDS):
    a.append(input("Enter a word: "))

for word in a:
    count = 0
    for letter in word:
        if letter == "w":
            count += 1
    if count >= 2:
        print(word)
```

34. Solution

```
number = int(input("Enter a number between 1 and 99: "))

digit1 = number // 10
digit2 = number % 10

number2romanOnes = {
    1: "I", 2: "II", 3: "III", 4: "IV", 5: "V", 6: "VI", 7: "VII", 8: "VIII", 9: "IX"
}

number2romanTens = {
    1: "X", 2: "XX", 3: "XXX", 4: "XL", 5: "L", 6: "LX", 7: "LXX", 8: "LXXX", 9: "XC"
}

roman = number2romanTens[digit1] + number2romanOnes[digit2]
print(roman)
```

Chapter 31

31.7 Review Questions: True/False

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

31.8 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

31.9 Review Exercises

1. Solution

Step	Statement	x	a
1	`a = [[None] * 3 for i in range(2)]`	?	? ? ? ? ? ?
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 = [[None] * 3 for i in range(2)]`	?	?	? ? ? ? ? ?
2	`i = 0`	**0**	?	? ? ? ? ? ?
3	`j = 0`	0	**0**	? ? ? ? ? ?
4	`a[i][j] = [i + 1] * 5 + j`	0	0	**5** ? ? ? ? ?
5	`j = 1`	0	**1**	5 ? ? ? ? ?
6	`a[i][j] = [i + 1] * 5 + j`	0	1	5 **6** ? ? ? ?
7	`j = 2`	0	**2**	5 6 ? ? ? ?
8	`a[i][j] = [i + 1] * 5 + j`	0	2	5 6 **7** ? ? ?
9	`i = 1`	**1**	2	5 6 7 ? ? ?

10	`j = 0`	1	**0**	5 6 7 ? ? ?
11	`a[i][j] = [i + 1] * 5 + j`	1	0	5 6 7 **10** ? ?
12	`j = 1`	1	**1**	5 6 7 10 ? ?
13	`a[i][j] = [i + 1] * 5 + j`	1	1	5 6 7 10 **11** ?
14	`j = 2`	1	**2**	5 6 7 10 11 ?
15	`a[i][j] = [i + 1] * 5 + j`	1	2	5 6 7 10 11 **12**

3. Solution

Step	Statement	i	j	a
1	`a = [[None] * 3 for i in range(3)]`	?	?	? ? ? ? ? ? ? ? ?
2	`j = 0`	?	**0**	? ? ? ? ? ? ? ? ?
3	`i = 0`	0	**0**	? ? ? ? ? ? ? ? ?
4	`a[i][j] = [i + 1] * 2 + j * 4`	0	0	**2** ? ? ? ? ? ? ? ?
5	`i = 1`	**1**	0	2 ? ? ? ? ? ? ? ?

6	`a[i][j] = [i + 1] * 2 + j * 4`	1	0	2 ? ? **4** ? ? ? ? ?
7	`i = 2`	**2**	0	2 ? ? 4 ? ? ? ? ?
8	`a[i][j] = [i + 1] * 2 + j * 4`	2	0	2 ? ? 4 ? ? **6** ? ?
9	`j = 1`	2	**1**	2 ? ? 4 ? ? 6 ? ?
10	`i = 0`	**0**	1	2 ? ? 4 ? ? 6 ? ?
11	`a[i][j] = [i + 1] * 2 + j * 4`	0	1	2 **6** ? 4 ? ? 6 ? ?
12	`i = 1`	**1**	1	2 6 ? 4 ? ? 6 ? ?
13	`a[i][j] = [i + 1] * 2 + j * 4`	1	1	2 6 ? 4 **8** ? 6 ? ?
14	`i = 2`	**2**	1	2 6 ? 4 8 ? 6 ? ?
15	`a[i][j] = [i + 1] * 2 + j * 4`	2	1	2 6 ? 4 8 ? 6 **10** ?
16	`j = 2`	2	**2**	2 6 ? 4 8 ? 6 10 ?

17	`i = 0`	**0**	2	2 6 ? 4 8 ? 6 10 ?
18	`a[i][j] = [i + 1] * 2 + j * 4`	0	2	2 6 **10** 4 8 ? 6 10 ?
19	`i = 1`	**1**	2	2 6 10 4 8 ? 6 10 ?
20	`a[i][j] = [i + 1] * 2 + j * 4`	1	2	2 6 10 4 8 **12** 6 10 ?
21	`i = 2`	**2**	2	2 6 10 4 8 12 6 10 ?
22	`a[i][j] = [i + 1] * 2 + j * 4`	2	2	2 6 10 4 8 12 6 10 **14**

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

```
ROWS = 10
COLUMNS = 15

a = [[None] * COLUMNS for i in range(ROWS)]
for i in range(ROWS):
    for j in range(COLUMNS):
        a[i][j] = int(input())

for i in range(ROWS):
    for j in range(COLUMNS):
        if a[i][j] % 2 != 0:
            print(i, ",", j)
```

10. Solution

```
ROWS = 10
COLUMNS = 6

a = [[None] * COLUMNS for i in range(ROWS)]
for i in range(ROWS):
    for j in range(COLUMNS):
        a[i][j] = float(input())

for i in range(ROWS):
    for j in range(0, COLUMNS, 2):
        print(a[i][j])
```

11. Solution

```
ROWS = 12
COLUMNS = 8

a = [[None] * COLUMNS for i in range(ROWS)]
for i in range(ROWS):
    for j in range(COLUMNS):
        a[i][j] = float(input())

total = 0
for i in range(1, ROWS, 2):
    for j in range(0, COLUMNS, 2):
        total += a[i][j]
print(total)
```

12. Solution

```
N = 8

a = [[None] * N for i in range(N)]
for i in range(N):
    for j in range(N):
        a[i][j] = float(input())

sumDiagonal = 0
sumAntidiagonal = 0
for k in range(N):
    sumDiagonal += a[k][k]
    sumAntidiagonal += a[k][N - k - 1]

print(sumDiagonal / N, sumAntidiagonal / N)
```

13. Solution

```
N = 5
```

```
a = [[None] * N for i in range(N)]
for i in range(N):
    for j in range(N):
        if i == N - j - 1:
            a[i][j] = 5
        elif i > N - j - 1:
            a[i][j] = 88
        else:
            a[i][j] = 11

for i in range(N):
    for j in range(N):
        print(a[i][j], end = "\t")
    print()
```

14. Solution

```
N = 5

a = [[None] * N for i in range(N)]
for i in range(N):
    for j in range(N):
        if i == N - j - 1:
            a[i][j] = 5
        elif i > N - j - 1:
            a[i][j] = 88
        else:
            a[i][j] = 11
        if i == j:
            a[i][j] = 0

for i in range(N):
    for j in range(N):
        print(a[i][j], end = "\t")
    print()
```

15. Solution

```
ROWS = 5
COLUMNS = 4

a = [[None] * COLUMNS for i in range(ROWS)]
for i in range(ROWS):
    for j in range(COLUMNS):
        a[i][j] = float(input())

for i in range(ROWS):
    for j in range(COLUMNS):
        if a[i][j] == int(a[i][j]):
```

```
            print(i, ",", j)
```

16. Solution

```
ROWS = 10
COLUMNS = 4

a = [[None] * COLUMNS for i in range(ROWS)]
for i in range(ROWS):
    for j in range(COLUMNS):
        a[i][j] = float(input())

count = 0
for i in range(ROWS):
    for j in range(COLUMNS):
        if a[i][j] < 0:
            count += 1

print(count)
```

17. Solution

```
ROWS = 3
COLUMNS = 4

a = [[None] * COLUMNS for i in range(ROWS)]
for i in range(ROWS):
    for j in range(COLUMNS):
        a[i][j] = input()

for i in range(ROWS):
    for j in range(COLUMNS):
        print(a[i][j], " ", end = "")
```

18. Solution

```
ROWS = 20
COLUMNS = 14

a = [[None] * COLUMNS for i in range(ROWS)]
for i in range(ROWS):
    for j in range(COLUMNS):
        a[i][j] = input()

for i in range(ROWS):
    for j in range(COLUMNS):
        if len(a[i][j]) < 5:
            print(a[i][j])
```

19. Solution

First approach

```
ROWS = 20
COLUMNS = 14

a = [[None] * COLUMNS for i in range(ROWS)]
for i in range(ROWS):
    for j in range(COLUMNS):
        a[i][j] = input()

lengthLimits = [5, 10, 20]

for k in range(3):
    for i in range(ROWS):
        for j in range(COLUMNS):
            if len(a[i][j]) < lengthLimits[k]:
                print(a[i][j])
```

Second approach

```
ROWS = 20
COLUMNS = 14

a = [[None] * COLUMNS for i in range(ROWS)]
for i in range(ROWS):
    for j in range(COLUMNS):
        a[i][j] = input()

for k in range(3):
    for i in range(ROWS):
        for j in range(COLUMNS):
            if len(a[i][j]) < 5 * 2 ** k:
                print(a[i][j])
```

Chapter 32

32.8 Review Questions: True/False

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

32.9 Review Questions: Multiple Choice

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

32.10 Review Exercises

1. Solution

```
STUDENTS = 15
TESTS = 5

grades = [[None] * TESTS for i in range(STUDENTS)]
for i in range(STUDENTS):
    for j in range(TESTS):
        grades[i][j] = int(input())

average = [None] * STUDENTS
for i in range(STUDENTS):
    average[i] = 0
    for j in range(TESTS):
        average[i] += grades[i][j]
    average[i] /= TESTS

for i in range(STUDENTS):
    print("Student No", (i + 1), ": ")

    if average[i] < 60:
        print("E/F")
    elif average[i] < 70:
        print("D")
```

```
    elif average[i] < 80:
        print("C")
    elif average[i] < 90:
        print("B")
    else:
        print("A")
```

2. Solution

First approach

```
OBJECTS = 5
FALLS = 10

g = [[None] * FALLS for i in range(OBJECTS)]
for i in range(OBJECTS):
    for j in range(FALLS):
        g[i][j] = int(input())

for i in range(OBJECTS):
    total = 0
    for j in range(FALLS):
        total += g[i][j]
    print("Average g for object No", (i + 1), ":", (total / FALLS))

for j in range(FALLS):
    total = 0
    for i in range(OBJECTS):
        total += g[i][j]
    print("Average g for fall No", (j + 1), ":", (total / OBJECTS))

total = 0
for i in range(OBJECTS):
    for j in range(FALLS):
        total += g[i][j]

print("Overall average g:", (total / (OBJECTS * FALLS)))
```

Second approach

```
from math import fsum
OBJECTS = 5
FALLS = 10

g = [[None] * FALLS for i in range(OBJECTS)]
for i in range(OBJECTS):
    for j in range(FALLS):
        g[i][j] = int(input())

for row in g:
    total = fsum(row)
    print("Average g for object No", (i + 1), ":", (total / FALLS))
```

```
for j in range(FALLS):
    total = 0
    for i in range(OBJECTS):
        total += g[i][j]
    print("Average g for fall No", (j + 1), ":", (total / OBJECTS))

total = 0
for row in g:
    total += fsum(row)
print("Overall average g:", (total / (OBJECTS * FALLS)))
```

3. Solution

First approach

```
PLAYERS = 15
MATCHES = 12

points = [[None] * MATCHES for i in range(PLAYERS)]
for i in range(PLAYERS):
    for j in range(MATCHES):
        points[i][j] = int(input())

for i in range(PLAYERS):
    total = 0
    for j in range(MATCHES):
        total += points[i][j]
    print("Total number of points for player No", (i + 1), ":", total)

for j in range(MATCHES):
    total = 0
    for i in range(PLAYERS):
        total += points[i][j]
    print("Total number of points for match No", (j + 1), ":", total)
```

Second approach

```
from math import fsum
PLAYERS = 15
MATCHES = 12

points = [[None] * MATCHES for i in range(PLAYERS)]
for i in range(PLAYERS):
    for j in range(MATCHES):
        points[i][j] = int(input())

for row in points:
    total = fsum(row)
    print("Total number of points for player No", (i + 1), ":", total)

for j in range(MATCHES):
```

```
    total = 0
    for i in range(PLAYERS):
        total += points[i][j]
    print("Total number of points for match No", (j + 1), ":", total)
```

4. Solution

```
CITIES = 20
HOURS = 24

temperatures = [[None] * HOURS for i in range(CITIES)]
for i in range(CITIES):
    for j in range(HOURS):
        temperatures[i][j] = float(input())

for j in range(HOURS):
    total = 0
    for i in range(CITIES):
        total += temperatures[i][j]
    if total / CITIES < 10:
        print("Hour:", (j + 1))
```

5. Solution

```
PLAYERS = 24
MATCHES = 10

names = [None] * PLAYERS
goals = [[None] * MATCHES for i in range(PLAYERS)]
for i in range(PLAYERS):
    names[i] = input()
    for j in range(MATCHES):
        goals[i][j] = int(input())

                                        # Or you can do the following:
for i in range(PLAYERS):                # for row in goals:
    total = 0                           #     print(names[i], ":", fsum(row) / MATCHES)
    for j in range(MATCHES):            #
        total += goals[i][j]            #
    print(names[i], ":", (total / MATCHES)) #

for j in range(MATCHES):
    total = 0
    for i in range(PLAYERS):
        total += goals[i][j]
    print("Match No", (j + 1), ":", total)
```

6. Solution

```
STUDENTS = 12
```

```
LESSONS = 6

names = [None] * STUDENTS
grades = [[None] * LESSONS for i in range(STUDENTS)]
for i in range(STUDENTS):
    names[i] = input()
    for j in range(LESSONS):
        grades[i][j] = int(input())

average = [None] * STUDENTS             # Or you can do the following:
for i in range(STUDENTS):               # for row in grades:
    total = 0                           #     average[i] = fsum(row) / LESSONS
    for j in range(LESSONS):            #     print(names[i], ":", average[i])
        total += grades[i][j]           #
    average[i] = total / LESSONS        #
    print(names[i], ":", average[i])    #

for j in range(LESSONS):
    total = 0
    for i in range(STUDENTS):
        total += grades[i][j]
    print(total / STUDENTS)

for i in range(STUDENTS):
    if average[i] < 60:
        print(names[i])

for i in range(STUDENTS):
    if average[i] > 89:
        print(names[i], "Bravo!")
```

7. Solution

```
ARTISTS = 15
JUDGES = 5

judgeNames = [None] * JUDGES
for j in range(JUDGES):
    judgeNames[j] = input("Enter name for judge No" + str(j + 1) + ": ")

artistNames = [None] * ARTISTS
songTitles = [None] * ARTISTS
score = [[None] * JUDGES for i in range(ARTISTS)]
for i in range(ARTISTS):
    artistNames[i] = input("Enter name for artist No" + str(i + 1) + ": ")
    songTitles[i] = input("Enter song title for artist " + artistNames[i] + ": ")
    for j in range(JUDGES):
        score[i][j] = int(input("Enter score for artist " + artistNames[i] + " gotten from judge " + \
                                judgeNames[j] + ": "))
```

```
for i in range(ARTISTS):
    total = 0
    for j in range(JUDGES):
        total += score[i][j]
    print(artistNames[i], ",", songTitles[i], ":", total)

for j in range(JUDGES):
    total = 0
    for i in range(ARTISTS):
        total += score[i][j]
    print(judgeNames[j], ":", total / ARTISTS)
```

8. Solution

```
PEOPLE = 30
MONTHS = 12

weights = [[None] * MONTHS for i in range(PEOPLE)]
heights = [[None] * MONTHS for i in range(PEOPLE)]
for i in range(PEOPLE):
    for j in range(MONTHS):
        weights[i][j] = int(input())
        heights[i][j] = int(input())

for i in range(PEOPLE):
    sumWeights = 0
    sumHeights = 0
    for j in range(MONTHS):
        sumWeights += weights[i][j]
        sumHeights += heights[i][j]
    averageWeight = sumWeights / MONTHS
    averageHeight = sumHeights / MONTHS
    print(averageWeight, ",", averageHeight)
    print(averageWeight * 702 / averageHeight ** 2)

for i in range(PEOPLE):
    print(weights[i][4] * 702 / heights[i][4] ** 2)
    print(weights[i][7] * 702 / heights[i][7] ** 2)
```

9. Solution

```
VAT = 0.19
CONSUMERS = 1000

meterReading = [[None] * 2 for i in range(CONSUMERS)]
for i in range(CONSUMERS):
    meterReading[i][0] = int(input())
    meterReading[i][1] = int(input())
```

```
total = 0
for i in range(CONSUMERS):
    consumed = meterReading[i][1] - meterReading[i][0]
    print(consumed)
    payment = consumed * 0.07
    payment += VAT * payment
    print(payment)

    total += consumed

print(total, total * 0.07 + total * 0.07 * VAT)
```

10. Solution

First approach

```
CURRENCIES = 4
DAYS = 5

usd = float(input("Enter an amount in US dollars: "))
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 in range(CURRENCIES):
    total = 0
    for j in range(DAYS):
        total += rate[i][j]
    average = total / DAYS
    print(usd, "US dollars =", usd / average, currency[i])
```

Second approach

```
from math import fsum
DAYS = 5

usd = float(input("Enter an amount in US dollars: "))
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 in range(len(rate)):
    average = fsum(rate[i]) / DAYS     #rate[i] returns the whole row
    print(usd, "US dollars =", usd / average, currency[i])
```

11. Solution

```
EMPLOYEES = 10
DAYS = 5

weekdays = ["Monday", "Tuesday", "Wednesday", "Thursday", "Friday"]

payRate = float(input())

names = [None] * EMPLOYEES
hoursWorkedPerDay = [[None] * DAYS for i in range(EMPLOYEES)]
for i in range(EMPLOYEES):
    names[i] = input()
    for j in range(DAYS):
        hoursWorkedPerDay[i][j] = int(input())

hoursWorkedPerWeek = [None] * EMPLOYEES
for i in range(EMPLOYEES):
    hoursWorkedPerWeek[i] = 0
    for j in range(DAYS):
        hoursWorkedPerWeek[i] += hoursWorkedPerDay[i][j]
    if hoursWorkedPerWeek[i] > 40:
        print(names[i])

totalGrossPay = 0
for i in range(EMPLOYEES):
    if hoursWorkedPerWeek[i] <= 40:
        grossPay = payRate * hoursWorkedPerWeek[i]
    else:
        grossPay = payRate * 40 + 1.5 * payRate * (hoursWorkedPerWeek[i] - 40)
    totalGrossPay += grossPay
    print(names[i], grossPay / 5)

print(totalGrossPay)

for i in range(EMPLOYEES):
    if hoursWorkedPerWeek[i] > 40:
        for j in range(DAYS):
            if hoursWorkedPerDay[i][j] > 8:
                print(names[i], weekdays[j], "Overtime!")

for j in range(DAYS):
    total = 0
    for i in range(EMPLOYEES):
        if hoursWorkedPerDay[i][j] <= 8:
            grossPay = payRate * hoursWorkedPerDay[i][j]
        else:
            grossPay = payRate * 8 + 1.5 * payRate * (hoursWorkedPerDay[i][j] - 8)
        total += grossPay
```

```
    print(weekdays[j], total)
```

12. Solution

First approach

```
ROWS = 3
COLUMNS = 4

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

b = [None] * (ROWS * COLUMNS)
k = 0
for i in range(ROWS):
    for j in range(COLUMNS):
        b[k] = a[i][j]
        k += 1

for k in range(len(b)):
    print(b[k], end = " ")
```

Second approach

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

b = []
for row in a:
    b = b + row

for element in b:
    print(element, end = " ")
```

13. Solution

```
ROWS = 3
COLUMNS = 3

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

b = [[None] * COLUMNS for i in range(ROWS)]
k = 0
for i in range(ROWS - 1, -1, -1):
    for j in range(COLUMNS):
        b[i][j] = a[k]
        k += 1
```

```
for i in range(ROWS):
    for j in range(COLUMNS):
        print(b[i][j], end = "\t")
    print()
```

Chapter 33

33.7 Review Questions: True/False

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

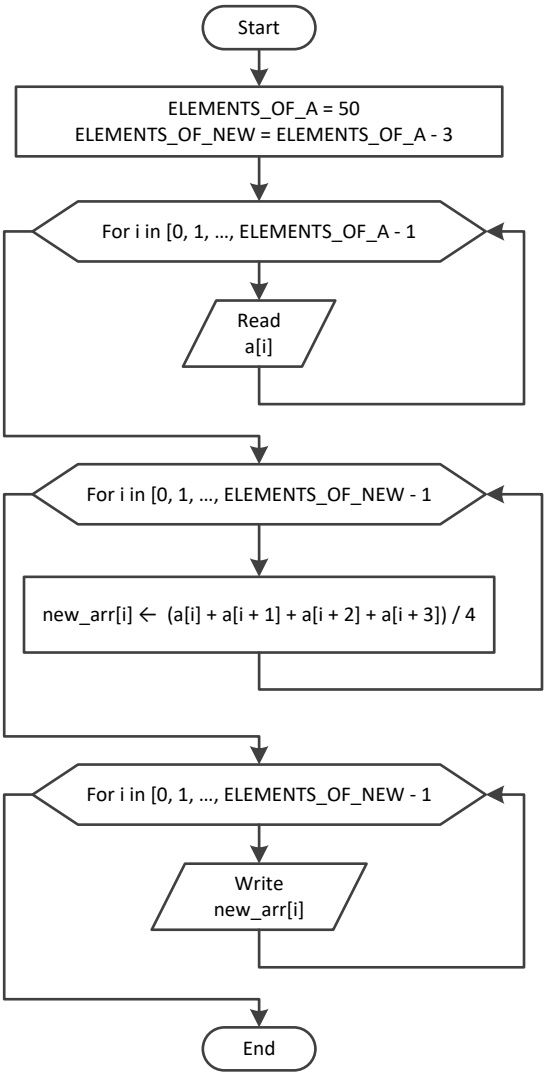
33.8 Review Exercises

1. Solution

```
for i in range(ROWS):
    for j in range(COLUMNS):
        a[i][j] = float(input())
        while a[i][j] == 0:
            print("Error")
            a[i][j] = float(input())
```

2. Solution

First approach



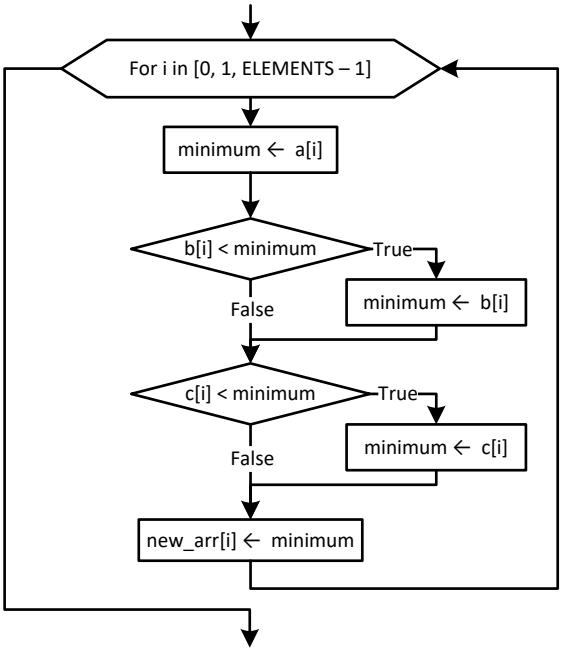

```
ELEMENTS_OF_A = 50
ELEMENTS_OF_NEW = ELEMENTS_OF_A - 3

a = [None] * ELEMENTS_OF_A
for i in range(ELEMENTS_OF_A):
    a[i] = float(input())

newArr = [None] * ELEMENTS_OF_NEW
for i in range(ELEMENTS_OF_NEW):
    newArr[i] = (a[i] + a[i + 1] + a[i + 2] + a[i + 3]) / 4

for i in range(ELEMENTS_OF_NEW):
    print(newArr[i], end = "\t")
```

Second approach

```
from math import fsum
ELEMENTS_OF_A = 50

a = [None] * ELEMENTS_OF_A

for i in range(ELEMENTS_OF_A):
    a[i] = float(input())

newArr = []
for i in range(ELEMENTS_OF_A - 3):
    newArr.append(fsum(a[i:i + 4]) / 4)

for element in newArr:
    print(element, end = "\t")
```

3. Solution

```
ELEMENTS = 15

a = [None] * ELEMENTS
for i in range(ELEMENTS):
    a[i] = float(input())

b = [None] * ELEMENTS
for i in range(ELEMENTS):
    b[i] = float(input())

c = [None] * ELEMENTS
for i in range(ELEMENTS):
    c[i] = float(input())

newArr = [None] * ELEMENTS
for i in range(ELEMENTS):       # Or you can do the following:
    minimum = a[i]              # newArr[i] = min(a[i], b[i], c[i])
    if b[i] < minimum:          #
        minimum = b[i]          #
    if c[i] < minimum:          #
        minimum = c[i]          #
    newArr[i] = minimum         #

for i in range(ELEMENTS):
    print(newArr[i])
```

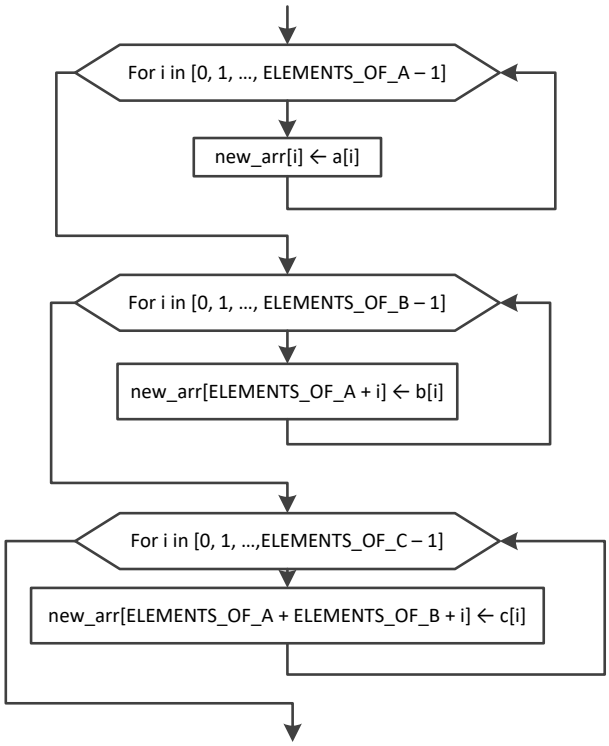


4. Solution

First approach

```
ELEMENTS_OF_A = 10
ELEMENTS_OF_B = 5
ELEMENTS_OF_C = 15
ELEMENTS_OF_NEW = ELEMENTS_OF_A + ELEMENTS_OF_B + ELEMENTS_OF_C

a = [None] * ELEMENTS_OF_A
for i in range(ELEMENTS_OF_A):
    a[i] = float(input())

b = [None] * ELEMENTS_OF_B
for i in range(ELEMENTS_OF_B):
    b[i] = float(input())

c = [None] * ELEMENTS_OF_C
for i in range(ELEMENTS_OF_C):
    c[i] = float(input())

newArr = [None] * ELEMENTS_OF_NEW
for i in range(ELEMENTS_OF_C):
    newArr[i] = c[i]
for i in range(ELEMENTS_OF_B):
    newArr[ELEMENTS_OF_C + i] = b[i]
for i in range(ELEMENTS_OF_A):
    newArr[ELEMENTS_OF_B + ELEMENTS_OF_C + i] = a[i]
```

```
for i in range(ELEMENTS_OF_NEW):
    print(newArr[i], end = "\t")
```



Second approach

```
ELEMENTS_OF_A = 10
ELEMENTS_OF_B = 5
ELEMENTS_OF_C = 15

a = [None] * ELEMENTS_OF_A
for i in range(ELEMENTS_OF_A):
    a[i] = float(input())

b = [None] * ELEMENTS_OF_B
for i in range(ELEMENTS_OF_B):
    b[i] = float(input())

c = [None] * ELEMENTS_OF_C
for i in range(ELEMENTS_OF_C):
    c[i] = float(input())

newArr = []
for element in c:
    newArr.append(element)
for element in b:
    newArr.append(element)
for element in a:
    newArr.append(element)
```

```
for element in newArr:
    print(element, end = "\t")
```

Third approach

```
ELEMENTS_OF_A = 10
ELEMENTS_OF_B = 5
ELEMENTS_OF_C = 15

a = [None] * ELEMENTS_OF_A
for i in range(ELEMENTS_OF_A):
    a[i] = float(input())

b = [None] * ELEMENTS_OF_B
for i in range(ELEMENTS_OF_B):
    b[i] = float(input())

c = [None] * ELEMENTS_OF_C
for i in range(ELEMENTS_OF_C):
    c[i] = float(input())

newArr = c + b + a

for element in newArr:
    print(element, end = "\t")
```

5. Solution

First approach

```
COLUMNS = 4
ROWS_OF_A = 3
ROWS_OF_B = 5
ROWS_OF_NEW = ROWS_OF_A + ROWS_OF_B

#Create lists a and b
a = [[10, 11, 12, 85], [3, 1, 5, 10], [-1, 2, -5, -10]]

b = [[10, 11, 16, 33], [11, 13, 5, 55], [-1, -2, -4, 44],
      [55, 33, 77, 12], [-110, 120, 132, 43]
    ]

#Create list newArr
newArr = [[None] * COLUMNS for i in range(ROWS_OF_NEW)]
for i in range(ROWS_OF_A):
    for j in range(COLUMNS):
        newArr[i][j] = a[i][j]
for i in range(ROWS_OF_B):
    for j in range(COLUMNS):
        newArr[ROWS_OF_A + i][j] = b[i][j]

#Display list newArr
```

```
for i in range(ROWS_OF_NEW):
    for j in range(COLUMNS):
        print(newArr[i][j], end = "\t")
    print()
```

Second approach

```
COLUMNS = 4
ROWS_OF_A = 3
ROWS_OF_B = 5
ROWS_OF_NEW = ROWS_OF_A + ROWS_OF_B

#Create lists a and b
a = [[10, 11, 12, 85], [3, 1, 5, 10], [-1, 2, -5, -10]]

b = [[10, 11, 16, 33], [11, 13, 5, 55], [-1, -2, -4, 44],
      [55, 33, 77, 12], [-110, 120, 132, 43]
    ]

#Create list newArr
newArr = []
for row in a:
    newArr.append(row)
for row in b:
    newArr.append(row)

#Display list newArr
for i in range(ROWS_OF_NEW):
    for j in range(COLUMNS):
        print(newArr[i][j], end = "\t")
    print()
```

Third approach

```
COLUMNS = 4
ROWS_OF_A = 3
ROWS_OF_B = 5
ROWS_OF_NEW = ROWS_OF_A + ROWS_OF_B

#Create lists a and b
a = [[10, 11, 12, 85], [3, 1, 5, 10], [-1, 2, -5, -10]]

b = [[10, 11, 16, 33], [11, 13, 5, 55], [-1, -2, -4, 44],
      [55, 33, 77, 12], [-110, 120, 132, 43]
    ]

#Create list newArr using the concatenation operator ( + )
newArr = a + b

#Display list newArr
for i in range(ROWS_OF_NEW):
    for j in range(COLUMNS):
        print(newArr[i][j], end = "\t")
    print()
```

6. Solution

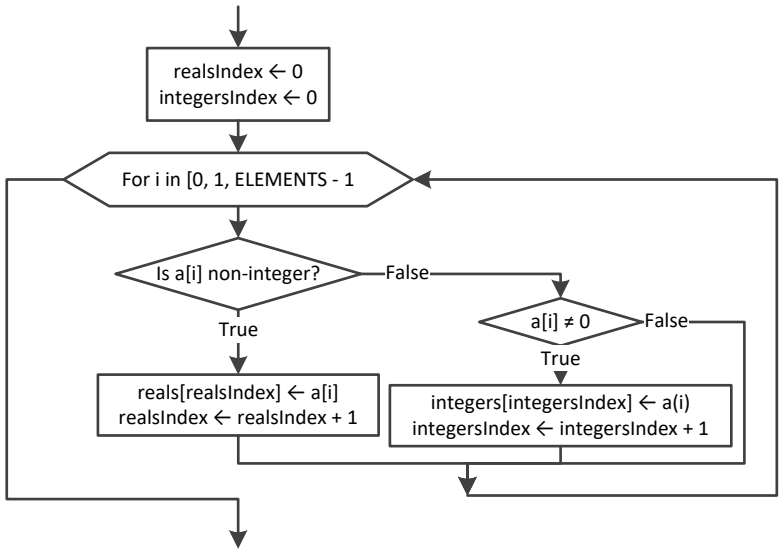
```
COLUMNS_OF_A = 10
COLUMNS_OF_B = 15
COLUMNS_OF_C = 20
ROWS = 5
COLUMNS = COLUMNS_OF_A + COLUMNS_OF_B + COLUMNS_OF_C

a = [[None] * COLUMNS_OF_A for i in range(ROWS)]
for i in range(ROWS):
    for j in range(COLUMNS_OF_A):
        a[i][j] = float(input())

b = [[None] * COLUMNS_OF_B for i in range(ROWS)]
for i in range(ROWS):
    for j in range(COLUMNS_OF_B):
        b[i][j] = float(input())

c = [[None] * COLUMNS_OF_C for i in range(ROWS)]
for i in range(ROWS):
    for j in range(COLUMNS_OF_C):
        c[i][j] = float(input())

newArr = [[None] * COLUMNS for i in range(ROWS)]
for i in range(ROWS):
    for j in range(COLUMNS_OF_A):
        newArr[i][j] = a[i][j]
for i in range(ROWS):
    for j in range(COLUMNS_OF_B):
        newArr[i][COLUMNS_OF_A + j] = b[i][j]
for i in range(ROWS):
    for j in range(COLUMNS_OF_C):
        newArr[i][COLUMNS_OF_A + COLUMNS_OF_B + j] = c[i][j]

for i in range(ROWS):
    for j in range(COLUMNS):
        print(newArr[i][j], end = "\t")
    print()
```

7. Solution

First approach

```
ELEMENTS = 50

a = [None] * ELEMENTS
for i in range(ELEMENTS):
    a[i] = float(input())

reals = [None] * ELEMENTS
```

```
integers = [None] * ELEMENTS
realsIndex = 0
integersIndex = 0
for i in range(ELEMENTS):
    if a[i] != int(a[i]):
        reals[realsIndex] = a[i]
        realsIndex += 1
    elif a[i] != 0:
        integers[integersIndex] = int(a[i])
        integersIndex += 1

for i in range(realsIndex):
    print(reals[i], end = "\t")

print()
for i in range(integersIndex):
    print(integers[i], end = "\t")
```



Second approach

```
ELEMENTS = 50

a = [None] * ELEMENTS

for i in range(ELEMENTS):
    a[i] = float(input())

reals = []
integers = []
for element in a:
    if element != int(element):
        reals.append(element)
    elif element != 0:
        integers.append(element)
```

```
for element in reals:
    print(element, end = "\t")

print()
for element in integers:
    print(element, end = "\t")
```

8. Solution

First approach

```
ELEMENTS = 50

a = [None] * ELEMENTS
for i in range(ELEMENTS):
    a[i] = int(input())

b = []
for element in a:
    digit1, r = divmod(element, 100)
    digit2, digit3 = divmod(r, 10)

    if digit1 < digit2 < digit3:
        b.append(element)

for element in b:
    print(element, end = "\t")
```

Second approach

```
ELEMENTS = 50

a = [None] * ELEMENTS
for i in range(ELEMENTS):
    a[i] = int(input())

b = []
for element in a:
    digit1, digit2, digit3 = str(element)    #digit1, digit2, digit3 are strings

    if digit1 < digit2 < digit3:             #Compare them as strings (no need to convert to integers)
        b.append(element)

for element in b:
    print(element, end = "\t")
```

9. Solution

```
PRODUCTS = 10
CITIZENS = 200
```

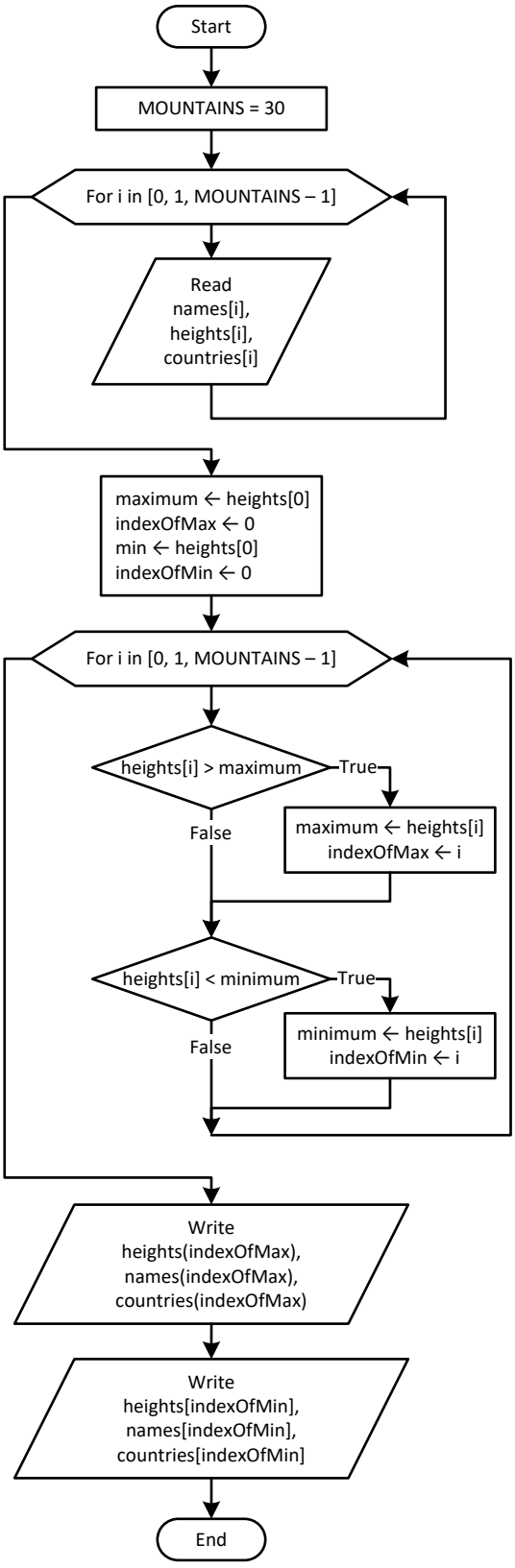
```
prodNames = [None] * PRODUCTS
answers = [[None] * CITIZENS for i in range(PRODUCTS)]
for i in range(PRODUCTS):
    prodNames[i] = input()
    for j in range(CITIZENS):
        answers[i][j] = input()
        while not("A" <= answers[i][j] <= "D"):
            print("Error! ")
            answers[i][j] = input()

countA = [None] * PRODUCTS
for i in range(PRODUCTS):
    countA[i] = 0
    for j in range(CITIZENS):
        if answers[i][j] == "A":
            countA[i] += 1
    print(prodNames[i], countA[i])

for j in range(CITIZENS):
    countB = 0
    for i in range(PRODUCTS):
        if answers[i][j] == "B":
            countB += 1
    print(countB)

maximum = countA[0]                # This code fragment can be replaced by the statement
for i in range(1, PRODUCTS):        # maximum = max(countA)
    if countA[i] > maximum:        #
        maximum = countA[i]        #

for i in range(PRODUCTS):
    if countA[i] == maximum:
        print(prodNames[i])
```

10. Solution

```
US_CITIES = 20
CANADIAN_CITIES = 20

usNames = [None] * US_CITIES
for i in range(US_CITIES):
    usNames[i] = input("Enter name for US city No" + str(i + 1) + ": ")

canadianNames = [None] * CANADIAN_CITIES
for j in range(CANADIAN_CITIES):
    canadianNames[j] = input("Enter name for Canadian city No" + str(j + 1) + ": ")

distances = [[None] * CANADIAN_CITIES for i in range(US_CITIES)]
for i in range(US_CITIES):
```

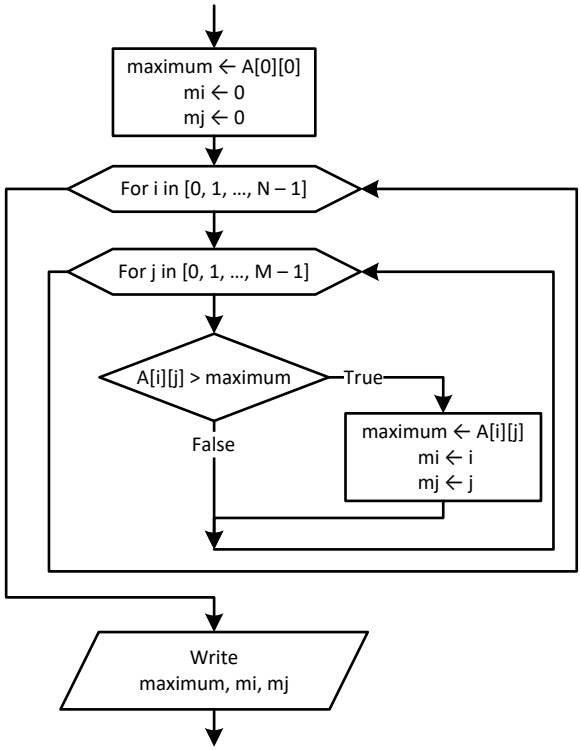
```
    for j in range(CANADIAN_CITIES):
        distances[i][j] = float(input("Enter distance between " + usNames[i] + " and " +        \
                                      canadianNames[j] + ": "))

for i in range(US_CITIES):
    minimum = distances[i][0]
    minJ = 0
    for j in range(1, CANADIAN_CITIES):
        if distances[i][j] < minimum:
            minimum = distances[i][j]
            minJ = j
    print("Closest Canadian city to", usNames[i], "is", canadianNames[minJ])
```

11. Solution


Start
MOUNTAINS = 30
For i in [0, 1, MOUNTAINS – 1]
Read
names[i],
heights[i],
countries[i]
maximum ← heights[0]
indexOfMax ← 0
min ← heights[0]
indexOfMin ← 0
For i in [0, 1, MOUNTAINS – 1]
heights[i] > maximum
True
False
maximum ← heights[i]
indexOfMax ← i
heights[i] < minimum
True
False
minimum ← heights[i]
indexOfMin ← i
Write
heights(indexOfMax),
names(indexOfMax),
countries(indexOfMax)
Write
heights[indexOfMin],
names[indexOfMin],
countries[indexOfMin]
End

```
MOUNTAINS = 30

names = [None] * MOUNTAINS
heights = [None] * MOUNTAINS
countries = [None] * MOUNTAINS
for i in range(MOUNTAINS):
    names[i] = input()
    heights[i] = float(input())
    countries[i] = input()

maximum = heights[0]
indexOfMax = 0
minimum = heights[0]
indexOfMin = 0
for i in range(1, MOUNTAINS):
    if heights[i] > maximum:
        maximum = heights[i]
        indexOfMax = i
    if heights[i] < minimum:
        minimum = heights[i]
        indexOfMin = i

print(heights[indexOfMax], names[indexOfMax], countries[indexOfMax])
print()
print(heights[indexOfMin], names[indexOfMin], countries[indexOfMin])
```

12. Solution



13. Solution

```
TEAMS = 26
GAMES = 15

names = [None] * TEAMS
results = [[None] * GAMES for i in range(TEAMS)]
for i in range(TEAMS):
    names[i] = input()
    for j in range(GAMES):
        results[i][j] = input()

points = [None] * TEAMS
for i in range(TEAMS):
    points[i] = 0
    for j in range(GAMES):
        if results[i][j] == "W":
            points[i] += 3
        elif results[i][j] == "T":
            points[i] += 1

maximum = points[0]
m_i = 0
```

```
for i in range(1, TEAMS):
    if points[i] > maximum:
        maximum = points[i]
        m_i = i

print(names[m_i])
```

14. Solution

First approach

```
OBJECTS = 10
FALLS = 20

heights = [[None] * FALLS for i in range(OBJECTS)]
times = [[None] * FALLS for i in range(OBJECTS)]
for i in range(OBJECTS):
    for j in range(FALLS):
        heights[i][j] = float(input())
        times[i][j] = float(input())

g = [[None] * FALLS for i in range(OBJECTS)]
for i in range(OBJECTS):
    for j in range(FALLS):
        g[i][j] = 2 * heights[i][j] / times[i][j] ** 2

minimum = [None] * OBJECTS
maximum = [None] * OBJECTS
for i in range(OBJECTS):
    minimum[i] = g[i][0]
    maximum[i] = g[i][0]
    for j in range(1, FALLS):
        if g[i][j] < minimum[i]:
            minimum[i] = g[i][j]
        if g[i][j] > maximum[i]:
            maximum[i] = g[i][j]

for i in range(OBJECTS):
    print(minimum[i], maximum[i])

maxim = maximum[0]
minim = minimum[0]
for i in range(1, OBJECTS):
    if maximum[i] > maxim:
        maxim = maximum[i]
    if minimum[i] < minim:
        minim = minimum[i]

print(minim, maxim)
```

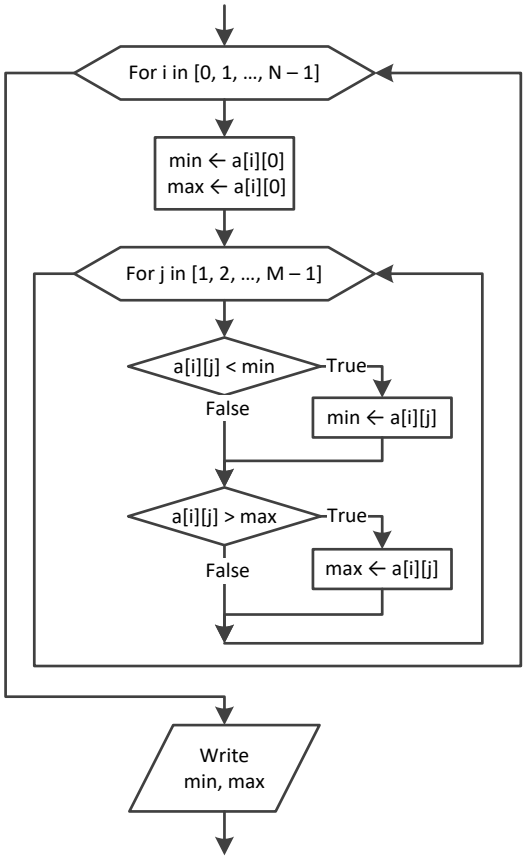
Second approach

```
OBJECTS = 10
FALLS = 20

g = [[None] * FALLS for i in range(OBJECTS)]
for i in range(OBJECTS):
    for j in range(FALLS):
        height = float(input())
        time = float(input())
        g[i][j] = 2 * height / time ** 2

minimum = [None] * OBJECTS
maximum = [None] * OBJECTS
for i in range(OBJECTS):
    minimum[i] = min(g[i])      # g[i] returns the whole row
    maximum[i] = max(g[i])

for i in range(OBJECTS):
    print(minimum[i], maximum[i])

print(min(minimum), max(maximum))
```

15. Solution

```
STATIONS = 10
DAYS = 365

names = [None] * STATIONS
co2 = [[None] * DAYS for i in range(STATIONS)]
for i in range(STATIONS):
    names[i] = input()
    for j in range(DAYS):
        co2[i][j] = float(input())

                                    # Or you can do the following:
average = [None] * STATIONS         # average = []
for i in range(STATIONS):           # for row in co2:
    average[i] = 0                  #     average.append(fsum(row) / DAYS)
    for j in range(DAYS):           #
        average[i] += co2[i][j]     #
    average[i] /= DAYS              #

minimum = average[0]
m_i = 0
for i in range(1, STATIONS):
    if average[i] < minimum:
        minimum = average[i]
        m_i = i
print(names[m_i])
```

16. Solution



17. Solution

First approach

```
ROWS = 20
COLUMNS = 30

b = [[None] * COLUMNS for i in range(ROWS)]

for i in range(ROWS):
    for j in range(COLUMNS):
        b[i][j] = float(input())

minimum = [None] * COLUMNS
maximum = [None] * COLUMNS

for j in range(COLUMNS):
    minimum[j] = b[0][j]
    maximum[j] = b[0][j]
    for i in range(1, ROWS):
        if b[i][j] < minimum[j]:
```

```
            minimum[j] = b[i][j]
        if b[i][j] > maximum[j]:
            maximum[j] = b[i][j]

for j in range(COLUMNS):
    print(minimum[j], maximum[j])
```

Second approach

```
ROWS = 20
COLUMNS = 30

b = [[None] * COLUMNS for i in range(ROWS)]

for i in range(ROWS):
    for j in range(COLUMNS):
        b[i][j] = float(input())

for j in range(COLUMNS):
    minimum = b[0][j]
    maximum = b[0][j]
    for i in range(1, ROWS):
        if b[i][j] < minimum:
            minimum = b[i][j]
        if b[i][j] > maximum:
            maximum = b[i][j]
    print(minimum, maximum)
```

18. Solution

```
TEAMS = 20
GAMES = 10

names = [None] * TEAMS
results = [[None] * GAMES for i in range(TEAMS)]
for i in range(TEAMS):
    names[i] = input("Enter team name: ")
    for j in range(GAMES):
        results[i][j] = input("Enter result for team " + names[i] + " for game No" + str(j + 1) + ": ")
        while results[i][j] not in ["W", "L", "T"]:
            results[i][j] = input("Error! Enter only value W, L, or T: ")

points = [None] * TEAMS
for i in range(TEAMS):
    points[i] = 0
    for j in range(GAMES):
        if results[i][j] == "W":
            points[i] += 3
        elif results[i][j] == "T":
            points[i] += 1
```

```
for m in range(3):    #Perfom only three passes
    swaps = False
    for n in range(TEAMS - 1, m, -1):
        if points[n] > points[n - 1]:
            points[n], points[n - 1] = points[n - 1], points[n]
            names[n], names[n - 1] = names[n - 1], names[n]
            swaps = True
    if not swaps: break

print("Gold:", names[0])
print("Silver:", names[1])
print("Bronze:", names[2])
```

19. Solution

```
PEOPLE = 50

names = [None] * PEOPLE
heights = [None] * PEOPLE
for i in range(PEOPLE):
    names[i] = input("Enter name for person No." + str(i + 1) + ": ")
    heights[i] = float(input("Enter height for person No." + str(i + 1) + ": "))

for m in range(PEOPLE - 1):
    for n in range(PEOPLE - 1, m, -1):
        if heights[n] > heights[n - 1]:
            heights[n], heights[n - 1] = heights[n - 1], heights[n]
            names[n], names[n - 1] = names[n - 1], names[n]
        elif heights[n] == heights[n - 1]:
            if names[n] < names[n - 1]:
                names[n], names[n - 1] = names[n - 1], names[n]

for i in range(PEOPLE):
    print(heights[i], names[i])
```

20. Solution

```
PEOPLE = 50

firstNames = [None] * PEOPLE
lastNames = [None] * PEOPLE
fatherNames = [None] * PEOPLE

for i in range(PEOPLE):
    firstNames[i] = input("Enter first name for person No." + str(i + 1) + ": ")
    lastNames[i] = input("Enter last name for person No." + str(i + 1) + ": ")
    fatherNames[i] = input("Enter father's name for person No." + str(i + 1) + ": ")

for m in range(PEOPLE - 1):
    for n in range(PEOPLE - 1, m, -1):
```

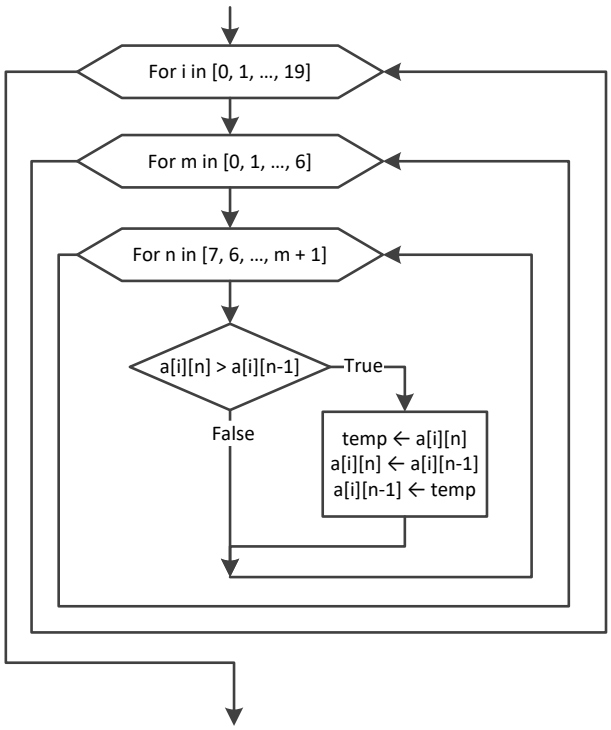
```
        if lastNames[n] > lastNames[n - 1]:
            lastNames[n], lastNames[n - 1] = lastNames[n - 1], lastNames[n]
            firstNames[n], firstNames[n - 1] = firstNames[n - 1], firstNames[n]
            fatherNames[n], fatherNames[n - 1] = fatherNames[n - 1], fatherNames[n]
        elif lastNames[n] == lastNames[n - 1]:
            if firstNames[n] < firstNames[n - 1]:
                firstNames[n], firstNames[n - 1] = firstNames[n - 1], firstNames[n]
                fatherNames[n], fatherNames[n - 1] = fatherNames[n - 1], fatherNames[n]
            elif firstNames[n] == firstNames[n - 1]:
                if fatherNames[n] < fatherNames[n - 1]:
                    fatherNames[n], fatherNames[n - 1] = fatherNames[n - 1], fatherNames[n]


for i in range(PEOPLE):
    print(lastNames[i], firstNames[i], fatherNames[i])
```

21. Solution

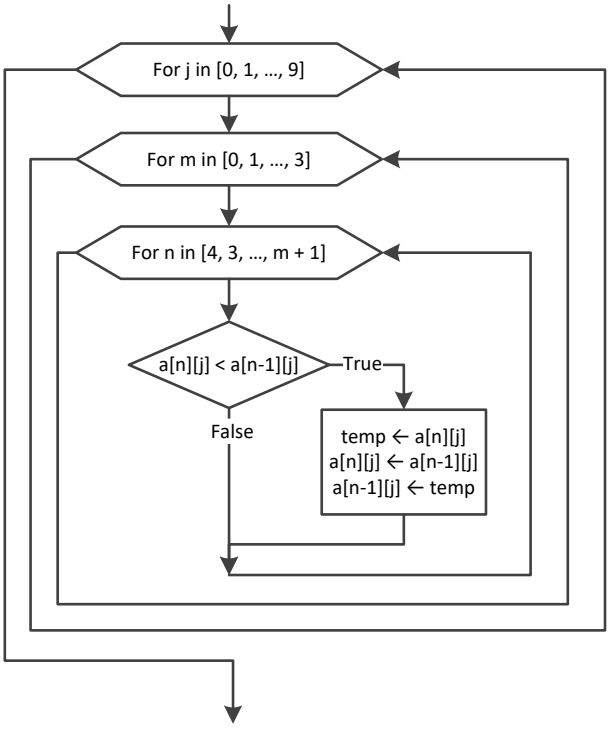
```
ARTISTS = 12
JUDGES = 10

artistNames = [None] * ARTISTS
score = [[None] * JUDGES for i in range(ARTISTS)]
for i in range(ARTISTS):
    artistNames[i] = input("Enter name for artist No." + str(i + 1) + ": ")
    for j in range(JUDGES):
        print("Enter score for artist: ", artistNames[i])
        score[i][j] = int(input(" gotten from judge No." + str(j + 1) + ": "))

total = [None] * ARTISTS            # Or you can do the following:
for i in range(ARTISTS):            # for row in score:
    total[i] = 0                    #     total[i] = fsum(row)
    for j in range(1, JUDGES):      #
        total[i] += score[i][j]     #

                                            # Or you can do the following:
for i in range(ARTISTS):                    # for i in range(ARTISTS):
    minimum = score[i][0]                   #     minimum = min(score[i])
    maximum = score[i][0]                   #     maximum = max(score[i])
    for j in range(1, JUDGES):              #     total[i] -= minimum - maximum
        if score[i][j] < minimum:           #     print(total[i])
            minimum = score[i][j]           #
        if score[i][j] > maximum:           #
            maximum = score[i][j]           #
    total[i] = total[i] - minimum - maximum #
    print(total[i])                         #

for m in range(ARTISTS - 1):
    for n in range(ARTISTS - 1, m, -1):
        if total[n] > total[n - 1]:
```

```
            total[n], total[n - 1] = total[n - 1], total[n]
            artistNames[n], artistNames[n - 1] = artistNames[n - 1], artistNames[n]
        elif total[n] == total[n - 1]:
            if artistNames[n] < artistNames[n - 1]:
                artistNames[n], artistNames[n - 1] = artistNames[n - 1], artistNames[n]

for i in range(ARTISTS):
    print(artistNames[i], total[i])
```

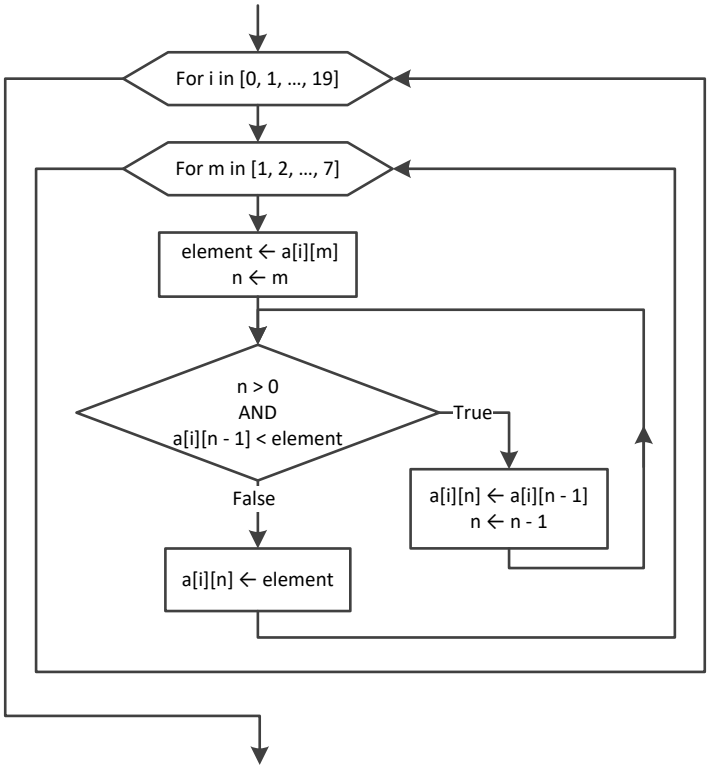
22. Solution



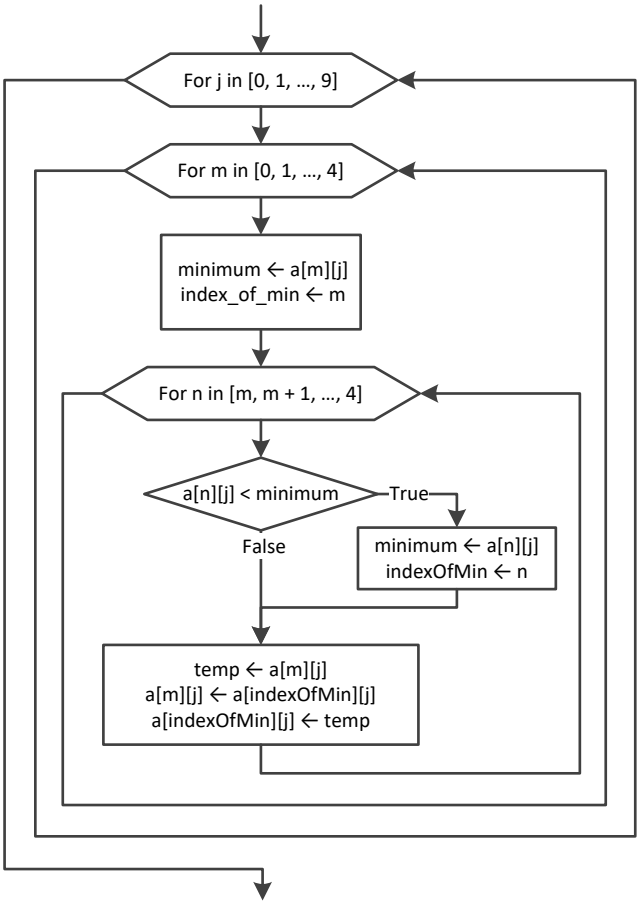
23. Solution


For j in [0, 1, ..., 9]
For m in [0, 1, ..., 3]
For n in [4, 3, ..., m + 1]
a[n][j] < a[n-1][j]
True
False
temp ← a[n][j]
a[n][j] ← a[n-1][j]
a[n-1][j] ← temp

24. Solution


For i in [0, 1, ..., 19]
For m in [1, 2, ..., 7]
element ← a[i][m]
n ← m
n > 0
AND
a[i][n - 1] < element
True
a[i][n] ← a[i][n - 1]
n ← n - 1
False
a[i][n] ← element

25. Solution



26. Solution

```
PEOPLE = 10
PUZZLES = 8

names = [None] * PEOPLE
times = [[None] * PUZZLES for i in range(PEOPLE)]
for i in range(PEOPLE):
    names[i] = input()
    for j in range(PUZZLES):
        hours = int(input())
        minutes = int(input())
        seconds = int(input())
        times[i][j] = hours * 3600 + minutes * 60 + seconds

for i in range(PEOPLE):
    for m in range(PUZZLES):
        minimum = times[i][m]
        indexOfMin = m
```

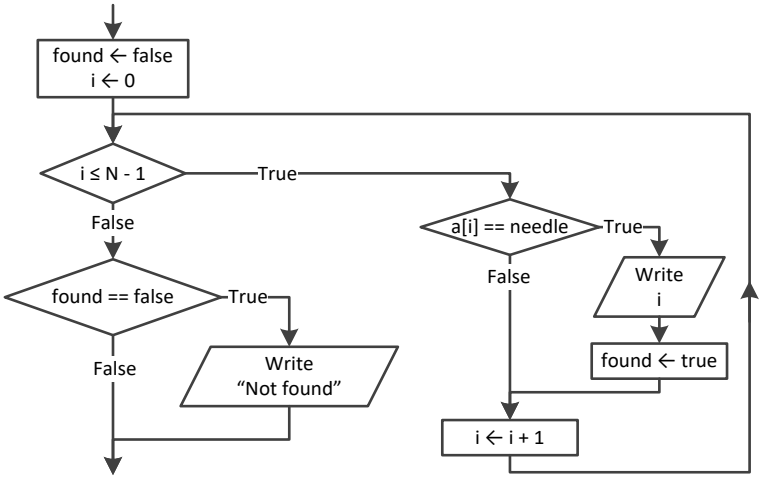
```
        for n in range(m, PUZZLES):
            if times[i][n] < minimum:
                minimum = times[i][n]
                indexOfMin = n
        times[i][m], times[i][indexOfMin] = times[i][indexOfMin], times[i][m]

for i in range(PEOPLE):
    print(names[i])
    for j in range(3):
        print(times[i][j])

                                    # Or you can do the following:
average = [None] * PEOPLE           # average = []
for i in range(PEOPLE):             # for row in times:
    average[i] = 0                  #     average.append(fsum(row) / PUZZLES)
    for j in range(PUZZLES):        #
        average[i] += times[i][j]   #
    average[i] /= PUZZLES           #

for m in range(3):  #Perform only 3 iterations
    minimum = average[m]
    indexOfMin = m
    for n in range(m, PEOPLE):
        if average[n] < minimum:
            minimum = average[n]
            indexOfMin = n
    average[m], average[indexOfMin] = average[indexOfMin], average[m]
    names[m], names[indexOfMin] = names[indexOfMin], names[m]

print(names[0], names[1], names[2])
```

27. Solution

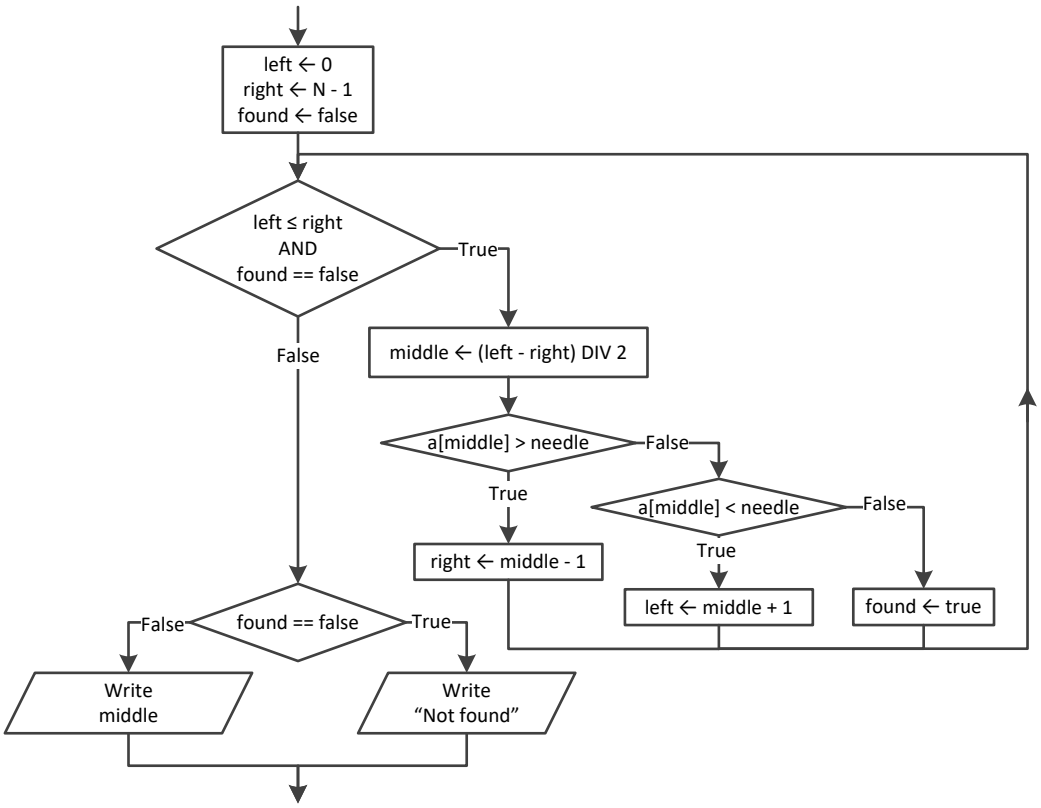
```
AREAS = 5
HOURS = 48

names = [None] * AREAS
CO2 = [[None] * HOURS for i in range(AREAS)]
for i in range(AREAS):
    names[i] = input()
    for j in range(HOURS):
        CO2[i][j] = float(input())
                                        # Or you can do the following:
averagePerHour = [None] * AREAS         # averagePerHour = []
for i in range(AREAS):                  # for row in CO2:
    averagePerHour[i] = 0               #     averagePerHour.append(fsum(row) / HOURS)
    for j in range(HOURS):              #
        averagePerHour[i] += CO2[i][j]  #
    averagePerhour[i] /= HOURS          #
```

```
for i in range(AREAS):
    print(names[i], averagePerHour[i])

averagePerCity = [None] * HOURS
for j in range(HOURS):
    averagePerCity[j] = 0
    for i in range(AREAS):
        averagePerCity[j] += CO2[i][j]
    averagePerCity[j] /= AREAS

for j in range(HOURS):
    print(averagePerCity[j])

maximum = averagePerCity[0]
m_j = 0
for j in range(1, HOURS):
    if averagePerCity[j] > max:
        maximum = averagePerCity[j]
        m_j = j
print(m_j)

maximum = CO2[0][0]
m_i = 0
m_j = 0
for i in range(AREAS):
    for j in range(HOURS):
        if CO2[i][j] > max:
            maximum = CO2[i][j]
            m_i = i
            m_j = j
print(m_j, names[m_i])

for m in range(1, AREAS):
    element1 = averagePerHour[m]
    element2 = names[m]

    n = m
    while n > 0 and averagePerHour[n - 1] < element1:
        averagePerHour[n] = averagePerHour[n - 1]
        names[n] = names[n - 1]
        n -= 1

    averagePerHour[n] = element1
    names[n] = element2

print(names[0], names[1],  names[2])
```

28. Solution



29. Solution



30. Solution

```
TEAMS = 20
WEEKS = 12
```

```
names = [None] * TEAMS
results = [[None] * WEEKS for i in range(TEAMS)]
for i in range(TEAMS):
    names[i] = input("Enter name for team No." + str(i + 1) + ": ")
    for j in range(WEEKS):
        results[i][j] = input("Enter result for week No." + str(j + 1) + " for " + \
                                                            names[i] + ": ")

#Get value to search and convert it to uppercase
needle = input("Enter a result to search: ").upper()

for i in range(TEAMS):
    found = False
    print("Found results for", names[i])
    for j in range(WEEKS):
        if results[i][j].upper() == needle:
            print("Week ", (j + 1))
            found = True

    if not found:
        print("No results!")
```

31. Solution

```
TEAMS = 10
GAMES = 16

names = [None] * TEAMS
goalsScored = [[None] * GAMES for i in range(TEAMS)]
goalsLetIn = [[None] * GAMES for i in range(TEAMS)]
for i in range(TEAMS):
    names[i] = input("Enter team name: ")
    for j in range(GAMES):
        goalsScored[i][j] = int(input("Enter goals scored: "))
        while goalsScored[i][j] < 0:
            goalsScored[i][j] = int(input("Error! Enter goals scored: "))

        goalsLetIn[i][j] = int(input("Enter goals let in: "))
        while goalsLetIn[i][j] < 0:
            goalsLetIn[i][j] = int(input("Error! Enter goals let in: "))

needle = input("Enter a team to search: ")

i = 0
while i < TEAMS - 1 and names[i] != needle:
    i += 1

if names[i] != needle:
    print("This team does not exist")
else:
```

```
    total = 0
    for j in range(GAMES):
        if goalsScored[i][j] > goalsLetIn[i][j]:
            total += 3
        elif goalsScored[i][j] == goalsLetIn[i][j]:
            total += 1
    print(total)
```

32. Solution

```
CLASS1 = 20
CLASS2 = 25

print("Class 1")
names1 = [None] * CLASS1
for i in range(CLASS1):
    names1[i] = input("Enter name: ")

print("Class 2")
names2 = [None] * CLASS2
for i in range(CLASS2):
    names2[i] = input("Enter name: ")

#Bubblesort
for m in range(CLASS1 - 1):
    for n in range(CLASS1 - 1, m, -1):
        if names1[n] < names1[n - 1]:
            names1[n], names1[n - 1] = names1[n - 1], names1[n]

for m in range(CLASS2 - 1):
    for n in range(CLASS2 - 1, m, -1):
        if names2[n] < names2[n - 1]:
            names2[n], names2[n - 1] = names2[n - 1], names2[n]

print("\nClass 1")
for i in range(CLASS1):
    print(names1[i])
print("\nClass 2")
for i in range(CLASS2):
    print(names2[i])

needle = input("Enter a name to search: ")

left = 0
right = CLASS1 - 1
found = False
while left <= right and not found:
    middle = (left + right) // 2
```

```
    if needle < names1[middle]:
        right = middle - 1
    elif needle > names1[middle]:
        left = middle + 1
    else:
        found = True

if found:
    print("Student found in Class No 1")
else:
    left = 0
    right = CLASS2 - 1
    while left <= right and not found:
        middle = (left + right) // 2

        if needle < names2[middle]:
            right = middle - 1
        elif needle > names2[middle]:
            left = middle + 1
        else:
            found = True

    if found:
        print("Student found in Class No 2")
    else:
        print("Student not found in either class")
```

33. Solution

```
usr = input("Enter username: ").upper()
pwd = input("Enter password: ").upper()

i = 0
while i < 99 and usernames[i].upper() != usr:
    i += 1

if usernames[i].upper() == usr and passwords[i].upper() == pwd:
    print("Login OK!")
else:
    print("Login Failed!")
```

34. Solution

```
valueStr = input("Enter a value to search: ")
found = False

for i in range(1000):
    if names[i] == valueStr:
        print(SSNs[i])
        found = True
```

```
if not found:
    value = int(valueStr)
    i = 0
    while i < 999 and SSNs[i] != value:
        i += 1

    if SSNs[i] == value:
        found = True
        print(names[i])

if not found:
    print("This value does not exist")
```

35. Solution

```
STUDENTS = 12
LESSONS = 6

grades = [[None] * LESSONS for i in range(STUDENTS)]
for i in range(STUDENTS):
    for j in range(LESSONS):
        while True:
            grades[i][j] = int(input())
            failure = False
            if grades[i][j] < 0:
                print("Error! You entered a negative value")
                failure = True
            elif grades[i][j] > 100:
                print("Error! You entered a value grater than 100")
                failure = True

            if not failure: break

                                    # Or you can do the following:
average = [None] * STUDENTS         # average = []
for i in range(STUDENTS):           # for row in grades:
    average[i] = 0                  #     average.append(fsum(row) / LESSONS)
    for j in range(LESSONS):        #
        average[i] += grades[i][j]  #
    average[i] /= LESSONS           #

found = False
for i in range(STUDENTS):
    if average[i] < 70:
        found = True
        break

if found:
```

```
    print("There is at least one student that has an average value below 70")
```

36. Solution

```
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" : "--..",
        " " : "/"
        }

word = input("Enter an English message: ")

for letter in word:
    print(morseAlphabet[letter.upper()], end = " ")
```

37. Solution

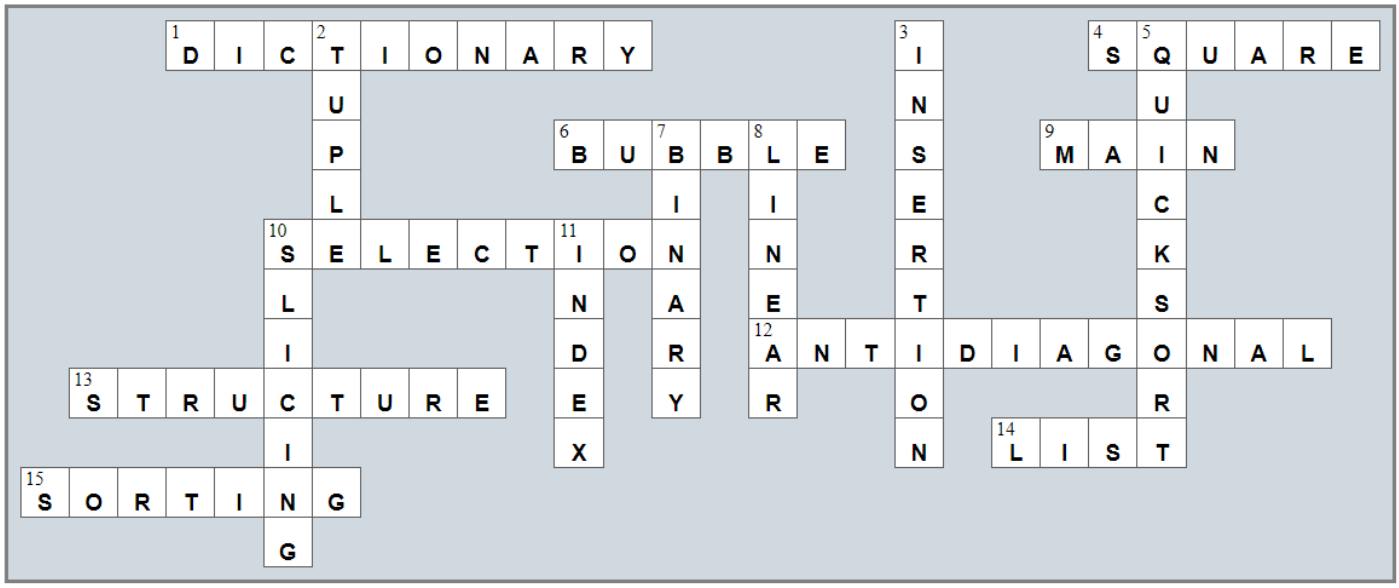
```
alphabet = "ABCDEFGHIJKLMNOPQRSTUVWXYZ";

#Create a dictionary to store the frequencies of each letter with initial
#frequencies all set to zero.
lettersFrequency  = {}
for letter in alphabet:
    lettersFrequency[letter]  = 0

sentence = input("Enter an English sentence: ")
```

```
#Iterate through the characters of the user-provided sentence and if it is a letter,
#update (increase) the corresponding frequency count in the lettersFrequency dictionary.
#Also count number of space characters and existing letters
countSpaces = 0
countExistingLetters = 0
for character in sentence.upper():
    if character in lettersFrequency:
        lettersFrequency[character] += 1
        countExistingLetters += 1
    elif character == " ":
        countSpaces += 1

#Display the frequency of each existing letter
for letter in lettersFrequency:
    if lettersFrequency[letter] > 0:
        print(letter + ":", lettersFrequency[letter])

#Count and display all non existing letters
countNonExistingLetters = 0
for letter in lettersFrequency:
    if lettersFrequency[letter] == 0:
        countNonExistingLetters += 1
        print(letter)

#Display percentage of letters that do not exist in relation to the letters of the English alphabet
print(countNonExistingLetters * 100 / 26, "%", sep = "")

#Display percentage of non-alphabetic characters in relation to the characters of
#the user-provided sentence (excluding space characters)
countUserProvidedCharacters = len(sentence) - countSpaces
countNonAlphabeticCharacters = countUserProvidedCharacters - countExistingLetters
print(countNonAlphabeticCharacters * 100 / countUserProvidedCharacters, "%", sep = "")
```

Review in “Data Structures in Python”

Review Crossword Puzzle

1.



Chapter 34

34.4 Review Questions: True/False

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

Chapter 35

35.8 Review Questions: True/False

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

35.9 Review Exercises

1. *Solution*

```
def findMax(a, b):
    if a > b:
        maximum = a
    else:
        maximum = b
    return maximum
```

2. *Solution*

Step	Statement	Main Code		Function sumDigits()		
		s	i	a	d1	d2
1	s = 0	**0**	?			
2	i = 25	0	**25**			
3	s += sumDigits(i)			**25**	?	?
4	d1 = a % 10			25	**5**	?
5	d2 = a // 10			25	5	**2**
6	return d1 + d2	**7**	25			
7	i = 26	7	**26**			

8	`s += sumDigits(i)`			**26**	?	?
9	`d1 = a % 10`			26	**6**	?
10	`d2 = a // 10`			26	6	**2**
11	`return d1 + d2`	**15**	26			
12	`i = 27`	15	**27**			
13	`s += sumDigits(i)`			**27**	?	?
14	`d1 = a % 10`			27	**7**	?
15	`d2 = a // 10`			27	7	**2**
16	`return d1 + d2`	**24**	27			
17	`print(s)`	It displays: 24				

3. Solution

Step	Statement	Main Code		Function 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`	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	`total += k`			2	**1**	1
13	`k = 2`			2	1	**2**
14	`total += k`			2	**3**	2
15	`return total`	**4**	2			
16	`i += 1`	4	**3**			
17	`while i < 6:`	True				
18	`if i % 2 == 1:`	True				
19	`s += 1`	**5**	3			
20	`i += 1`	5	**4**			
21	`while i < 6:`	True				
22	`if i % 2 == 1:`	False				

23	`s += sss(i)`			**4**	?	?
24	`total = 0`			4	**0**	?
25	`k = 1`			4	0	**1**
26	`total += k`			4	**1**	1
27	`k = 2`			4	1	**2**
28	`total += k`			4	**3**	2
29	`k = 3`			4	3	**3**
30	`total += k`			4	**6**	4
31	`k = 4`			4	6	**4**
32	`total += k`			4	**10**	4
33	`return total`	**15**	4			
34	`i += 1`	15	**5**			
35	`while i < 6:`	True				
36	`if i % 2 == 1:`	True				
37	`s += 1`	**16**	5			
38	`i += 1`	16	**6**			
39	`while i < 6:`	False				
40	`print(s)`	It displays: 16				

4. Solution

Step	Statement	Main Code				Function `customDiv()`	
		k	**m**	**a**	**x**	**b**	**d**
1	`k = int(input())`	**12**	?	?	?		
2	`m = 2`	12	**2**	?	?		
3	`a = 1`	12	2	**1**	?		
4	`while a < 6:`	True					
5	`if k % m != 0:`	False					
6	`x = a + m + customDiv(m, a)`					2	1
7	`return (b + d) // 2`	12	2	1	**4**		
8	`print(m, a, x)`	It displays: 2 1 4					
9	`a += 2`	12	2	**3**	4		
10	`m += 1`	12	**3**	3	4		
11	`while a < 6:`	True					
12	`if k % m != 0:`	False					
13	`x = a + m + customDiv(m, a)`					3	3

14	`return (b + d) // 2`	12	3	3	**9**		
15	`print(m, a, x)`	It displays: 3 3 9					
16	`a += 2`	12	3	**5**	9		
17	`m += 1`	12	**4**	5	9		
18	`while a < 6:`	True					
19	`if k % m != 0:`	False					
20	`x = a + m + customDiv(m, a)`					4	5
21	`return (b + d) // 2`	12	4	5	**13**		
22	`print(m, a, x)`	It displays: 4 5 13					
23	`a += 2`	12	4	**7**	13		
24	`m += 1`	12	**5**	7	13		
25	`while a < 6:`	False					

5. Solution

Step	Statement	Main Code		Function `display()`
		i	x	a
1	`i = 0`	**0**	?	
2	`x = int(input())`	0	**3**	
3	`display(x)`			**3**
4	`if a % 2 == 0:`			False
5	`print(a, "is odd")`	It displays: 3 is odd		
6	`i = 1`	**1**	3	
7	`x = int(input())`	1	**7**	
8	`display(x)`			**7**
9	`if a % 2 == 0:`			False
10	`print(a, "is odd")`	It displays: 7 is odd		
11	`i = 2`	**2**	7	
12	`x = int(input())`	2	**9**	
13	`display(x)`			**9**
14	`if a % 2 == 0:`			False
15	`print(a, "is odd")`	It displays: 9 is odd		
16	`i = 3`	**3**	9	
17	`x = int(input())`	3	**2**	
18	`display(x)`			**2**
19	`if a % 2 == 0:`			True

20	`print(a + " is even")`	It displays: 2 is even		
21	`i = 4`	**4**	2	
22	`x = int(input())`	4	**4**	
23	`display(x)`			**4**
24	`if a % 2 == 0:`			True
25	`print(a, "is even")`	It displays: 4 is even		

6. *Solution*

Step	Statement	Main Code		Function `division()`	
		x	y	a	b
1	`x = 20`	**20**	?		
2	`y = 30`	20	**30**		
3	`while x % y < 30:`	True			
4	`division(y, x)`			**30**	**20**
5	`b = b // a`			30	**0**
6	`print(a * b)`	It displays: 0			
7	`x = 4 * y`	**120**	30		
8	`y += 1`	120	**31**		
9	`while x % y < 30:`	True			
10	`division(y, x)`			**31**	**120**
11	`b = b // a`			31	**3**
12	`print(a * b)`	It displays: 93			
13	`x = 4 * y`	**124**	31		
14	`y += 1`	124	**32**		
15	`while x % y < 30:`	True			
16	`division(y, x)`			**32**	**124**
17	`b = b // a`			32	**3**
18	`print(a * b)`	It displays: 96			
19	`x = 4 * y`	**128**	32		
20	`y += 1`	128	**33**		
21	`while x % y < 30:`	True			
22	`division(y, x)`			**33**	**128**
23	`b = b // a`			33	**3**
24	`print(a * b)`	It displays: 99			
25	`x = 4 * y`	**132**	33		

26	`y += 1`	132	**34**		
27	`while x % y < 30:`	False			

7. Solution

Step	Statement	Main Code		Function `calculate()`		
		i	m	n	s	j
1	`i = 0`	**0**	?			
2	`m = int(input())`	0	**2**			
3	`calculate(m)`			**2**	?	?
4	`s = 0`			2	**0**	?
5	`j = 2`			2	0	**2**
6	`s = s + j ** 2`			2	**4**	2
7	`j = 4`			2	4	**4**
8	`s = s + j ** 2`			2	**20**	4
9	`print(s)`	It displays: 20				
10	`i = 1`	**1**	2			
11	`m = int(input())`	1	**3**			
12	`calculate(m)`			**3**	?	?
13	`s = 0`			3	**0**	?
14	`j = 2`			3	0	**2**
15	`s = s + j ** 2`			3	**4**	2
16	`j = 4`			3	4	**4**
17	`s = s + j ** 2`			3	**20**	4
18	`j = 6`			3	20	**6**
19	`s = s + j ** 2`			3	**56**	6
20	`print(s)`	It displays: 56				
21	`i = 2`	**2**	3			
22	`m = int(input())`	2	**4**			
23	`calculate(m)`			**4**	?	?
24	`s = 0`			4	**0**	?
25	`j = 2`			4	0	**2**
26	`s = s + j ** 2`			4	**4**	2
27	`j = 4`			4	4	**4**
28	`s = s + j ** 2`			4	**20**	4
29	`j = 6`			4	20	**6**

30	`s = s + j ** 2`			4	**56**	6
31	`j = 8`			4	56	**8**
32	`s = s + j ** 2`			4	**120**	8
33	`print(s)`	It displays: 120				

8. Solution

```
def findSum(a, b, c):
    return a + b + c
```

9. Solution

```
def findAvg(a, b, c, d):
    return (a + b + c + d) / 4
```

10. Solution

```
def maximum(a, b, c):
    m = a
    if b > m:
        m = b
    if c > m:
        m = c

    return m
```

11. Solution

First approach

```
def displayMax(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
    print(m)
```

Second approach

```
def displayMax(a, b, c, d, e):
    print(max(a, b, c, d, e))
```

12. Solution

```
def myRound(x):
    digitToCheck = int(x * 1000) % 10
    if digitToCheck >= 5:
```

```
        returnValue = (int(x * 100) + 1) / 100.0
    else:
        returnValue = (int(x * 100)) / 100.0

    return returnValue
```

13. Solution

```
def findMin(a, b):
    minimum = a
    if b < minimum:
        minimum = b
    return minimum

#Main code starts here
print("Enter four numbers: ")
x1 = float(input())
x2 = float(input())
x3 = float(input())
x4 = float(input())

                                # Or you can do the following
temp1 = findMin(x1, x2)         #
temp2 = findMin(x3, x4)         #
print(findMin(temp1, temp2))    # print(findMin(findMin(x1, x2), findMin(x3, x4)))
```

14. Solution

```
def KelvinToFahrenheit(kelvin):
    return 1.8 * kelvin - 459.67

def KelvinToCelsius(kelvin):
    return kelvin - 273.15

#Main code starts here
k = float(input("Enter a temperature in degrees Kelvin: "))
print("Fahrenheit:", KelvinToFahrenheit(k))
print("Celsius:", KelvinToCelsius(k))
```

15. Solution

```
def bmi(w, h):
    b = w * 703 / h ** 2
    if b < 16:
        returnValue = "You must add weight."
    elif b < 18.5:
        returnValue = "You should add some weight."
    elif b < 25:
        returnValue = "Maintain your weight."
    elif b < 30:
```

```
        returnValue = "You should lose some weight."
    else:
        returnValue = "You must lose weight."

    return returnValue

#Main code starts here
weight = float(input("Enter your weight (in pounds): "))
while weight < 0:
    weight = float(input("Error! Enter your weight (in pounds): "))

age = int(input("Enter your age: "))
while age < 18:
    age = int(input("Error! Enter your age: "))

height = float(input("Enter your height (in inches): "))
while height < 0:
    height = float(input("Error! Enter your height (in inches): "))

print(bmi(weight, height))
```

16. Solution

```
def numOfDays(year, month):
    if month in [4, 6, 9, 11]:
        days = 30
    elif month == 2:
        if year % 4 == 0 and year % 100 != 0 or year % 400 == 0:
            days = 29
        else:
            days = 28
    else:
        days = 31
    print(days)

#Main code starts here
y = int(input("Enter a year: "))
for m in range(1, 13):
    numOfDays(y, m)
```

17. Solution

```
def numOfDays(year, month):
    if month in [4, 6, 9, 11]:
        days = 30
    elif month == 2:
        if year % 4 == 0 and year % 100 != 0 or year % 400 == 0:
            days = 29
        else:
            days = 28
```

```
    else:
        days = 31

    return days

#Main code starts here
y = int(input("Enter a year: "))
m1 = int(input("Enter a month: "))
m2 = int(input("Enter a second month: "))

total = 0
for m in range(m1, m2 + 1):
    total += numOfDays(y, m)

print(total)
```

18. Solution

```
def displayMenu():
    print()
    print("1. Convert meters to miles")
    print("2. Convert miles to meters")
    print("3. Exit")
    print("Enter a choice: ", end = "")

def metersToMiles(meters):
    print(meters, "meters equals", (meters / 1609.344), "miles")

def milesToMeters(miles):
    print(miles, "miles equals", (miles * 1609.344), "meters")

#Main code starts here
displayMenu()
choice = int(input())
while choice != 3:
    distance = float(input("Enter distance: "))
    if choice == 1:
        metersToMiles(distance)
    else:
        milesToMeters(distance)

    displayMenu()
    choice = int(input())
```

19. Solution

```
def amountToPay(seconds):
    if seconds <= 600:
        extra = 0
    elif seconds <= 1200:
        extra = (seconds - 600) * 0.01
```

```
    else:
        extra = 600 * 0.01 + (seconds - 1200) * 0.02

    totalWithoutTax = 10 + extra
    tax = totalWithoutTax * 11 / 100
    total = totalWithoutTax + tax

    print("Total amount to pay:", total)

#Main code starts here
seconds = int(input("Enter number of seconds: "))
amountToPay(seconds)
```

Chapter 36

36.9 Review Questions: True/False

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

36.10 Review Exercises

1. Solution

It displays: 5

2. Solution

It displays: 14

3. Solution

It displays: 14

4. Solution

It displays: hellohellohello

5. Solution

It displays: 15

6. Solution

It displays: 11 4

7. Solution

It displays: 3

8. Solution

Within the function `getNumOfDigits()`, the corresponding element eventually becomes 0, and since the list `val` is passed to the function by reference, that zero also reflects back to the main code.

To resolve this issue, all you have to do is assign the value of the corresponding element to an auxiliary variable and let this variable become zero.

```
ELEMENTS = 5

def getNumOfDigits(x, index):
    count = 0

    auxVar = x[index]

    while auxVar != 0:
        count += 1
        auxVar = auxVar // 10
    return count

#Main code starts here
val = [None] * ELEMENTS

for i in range(ELEMENTS):
    val[i] = int(input())

for i in range(ELEMENTS):
    print(getNumOfDigits(val, i), "digits in number", val[i])
```

9. Solution

```
STUDENTS = 10
LESSONS = 5

def part1(names, grades):
    for i in range(STUDENTS):
        names[i] = input("Enter name No." + str(i + 1) + ": ")
        for j in range(LESSONS):
            grades[i][j] = int(input("Enter grade for lesson No." + str(j + 1) + ": "))

def part2(grades):
    average = [None] * STUDENTS

    for i in range(STUDENTS):
        average[i] = 0
        for j in range(LESSONS):
            average[i] += grades[i][j]
        average[i] /= LESSONS
    return average

def part3(average, names):
    for m in range(1, STUDENTS):
```

```
        for n in range(STUDENTS - 1, m - 1, -1):
            if average[n] > average[n - 1]:
                average[n], average[n - 1] = average[n - 1], average[n]
                names[n], names[n - 1] = names[n - 1], names[n]
            elif average[n] == average[n - 1]:
                if names[n].CompareTo(names[n - 1]) < 0:
                    names[n], names[n - 1] = names[n - 1], names[n]

#Main code starts here
names = [None] * STUDENTS
grades = [[None] * LESSONS for i in range(STUDENTS)]

part1(names, grades)
average = part2(grades)
part3(average, names)

for i in range(STUDENTS):
    print(names[i], "\t", average[i])
```

10. Solution

```
def part1():
    message = input("Enter a message: ").lower()
    return message

def part2(message):
    messageClean = ""
    for i in range(len(message)):
        if message[i] not in " ,.?":
            messageClean += message[i]
    return messageClean

def part3(messageClean):
    middlePos = (len(messageClean) - 1) // 2
    j = len(messageClean) - 1
    palindrome = True
    for i in range(middlePos + 1):
        if messageClean[i] != messageClean[j]:
            palindrome = False
            break
        j -= 1
    return palindrome

def part4(message):
    messageClean = part2(message)
    palindrome = part3(messageClean)
    return palindrome

#Main code starts here
```

```
message = part1()
palindrome = part4(message)
if palindrome:
    print("The message is palindrome")
```

11. Solution

```
a = int(input())
b = int(input())
c = int(input())
d = int(input())

maximum = a
if b > maximum:
    maximum = b
if c > maximum:
    maximum = c
if d > maximum:
    maximum = d

print(maximum)
```

12. Solution

First approach

```
def f1(a, b, c):
    total = a + b + c      #Or you can do
    average = total / 3    #the following:
    return total, average  #return a + b + c, (a + b + c) / 3
```

Second approach

```
def f1(a, b, c, returningList):
    total = a + b + c                  #Or you can do
    average = total / 3                #the following:
    returningList = [total, average]   #return [a + b + c, (a + b + c) / 3]
```

13. Solution

```
def myRound(x, decimalPlaces = 2):
    digitToCheck = x * 10 ** (decimalPlaces + 1) % 10
    if digitToCheck >= 5:
        returnValue = int(x * 10 ** decimalPlaces + 1) / 10 ** decimalPlaces
    else:
        returnValue = int(x * 10 ** decimalPlaces) / 10 ** decimalPlaces
    return returnValue
```

14. Solution

```
def getInput():
    while True:
        answer = input("Enter Yes or No: ").upper()
```

```
        if answer == "YES" or answer == "NO": break

    return answer == "YES"  #This returns True or False

def findArea(b, h):
    return b * h

#Main code starts here
while True:
    b = float(input("Enter the base of the parallelogram: "))
    h = float(input("Enter the height of the parallelogram: "))

    print("Area =", findArea(b, h))

    print("Would you like to repeat? ")
    if getInput() == False: break  #Or you can write: if not getInput(): break
```

15. Solution

```
STUDENTS = 100

def getLists(names, grades):
    for i in range(STUDENTS):
        names[i] = input("Enter name: ")
        grades[i] = int(input("Enter grade: "))

def getAverage(grades):
    total = 0                   # Or you can do the following:
    for i in range(STUDENTS):   #
        total += grades[i]      #
    return total / STUDENTS     # return fsum(grades) / STUDENTS

def sortLists(grades, names):
    for m in range(1, STUDENTS):
        elementGrds = grades[m]
        elementNms = names[m]

        n = m
        while n > 0 and grades[n - 1] > elementGrds:
            grades[n] = grades[n - 1]
            names[n] = names[n - 1]
            n -= 1

        grades[n] = elementGrds
        names[n] = elementNms

#Main code starts here
names = [None] * STUDENTS
grades = [None] * STUDENTS
```

```
getLists(names, grades)
average = getAverage(grades)
sortLists(grades, names)
for i in range(STUDENTS):
    if grades[i] < average:
        print(names[i])
```

16. Solution

First approach

```
JUDGES = 10

def getList():
    score = [None] * JUDGES
    for i in range(JUDGES):
        score[i] = int(input("Judge No" + str(i + 1) + ". Enter score: "))
    return score

def findMinMax(score):
    minimum = score[0]
    maximum = score[0]
    for i in range(1, JUDGES):
        if score[i] > maximum:
            maximum = score[i]
        if score[i] < minimum:
            minimum = score[i]

    return minimum, maximum

#Main code starts here
name = input("Enter artist's name: ")
score = getList()
minimum, maximum = findMinMax(score)

total = 0
for i in range(JUDGES):
    total += score[i]

points = total - minimum - maximum
print("Artist", name, "got", points, "points")
```

Second approach

```
from math import fsum
JUDGES = 10

def getList():
    score = [None] * JUDGES
    for i in range(JUDGES):
        score[i] = int(input("Judge No" + str(i + 1) + ". Enter score: "))
    return score
```

```
def findMinMax(score):
    return min(score), max(score)

#Main code starts here
name = input("Enter artist's name: ")
score = getList()
minimum, maximum = findMinMax(score)

points = fsum(score) - minimum - maximum
print("Artist", name, "got", points, "points")
```

17. Solution

```
def sumRecursive(n):
    if n == 1:
        return 1
    else:
        return sumRecursive(n - 1) + n

#Main code starts here
num = int(input())
print(sumRecursive(num))
```

18. Solution

```
def woc(index):
    if index == 1:
        return 1
    else:
        return 2 * woc(index - 1)

#Main code starts here
total = 0
for i in range(1, 65):
    total += woc(i)
print(total)
```

19. Solution

```
def fib(n):
    if n <= 1:
        return n
    else:
        return fib(n - 1) + fib(n - 2)

#Main code starts here
num = int(input())
print(fib(num - 1))
```

20. Solution

```
def tribonacci(n):
    if n == 0:
        return 0
    elif n == 1 or n == 2:
        return 1
    else:
        return tribonacci(n - 1) + tribonacci(n - 2) + tribonacci(n - 3)
```

21. Solution

```
def myPow(n,p):
    if p == 0:
        return 1
    elif p < 0:
        return 1 / (n * myPow(n, -p - 1))
    else:
        return n * myPow(n, p - 1)
```

22. Solution

```
from math import pi

def factorial(value):
    if value == 1:
        return 1
    else:
        return value * factorial(value - 1)

def myCos(x, i = 40):
    if i == 0:
        return 1
    else:
        return myCos(x, i - 4) + x ** i / factorial(i) - x ** (i - 2) / factorial(i - 2)

#Main code starts here
print(myCos(pi / 4))
```

Chapter 37

37.3 Review Exercises

1. Solution

```
def displayMenu():
    print("-----------------------------------------------")
    print("1. Convert USD to Euro (EUR)")
    print("2. Convert USD to British Pound Sterling (GBP)")
    print("3. Convert USD to Japanese Yen (JPY)")
    print("4. Convert USD to Canadian Dollar (CAD)")
    print("5. Exit")
    print("-----------------------------------------------")
    print("Enter a choice: ", end = "")

def USD_to_EU(value):
    return value * 0.94

def USD_to_GBP(value):
    return value * 0.81

def USD_to_JPY(value):
    return value * 149.11

def USD_to_CAD(value):
    return value * 1.36

#Main code starts here
displayMenu()
choice = int(input())
while choice != 5:
    amount = float(input("Enter an amount in US dollars: "))
    if choice == 1:
        print(amount, "USD =", USD_to_EU(amount), "Euro")
    elif choice == 2:
        print(amount, "USD =", USD_to_GBP(amount), "GBP")
    elif choice == 3:
        print(amount, "USD =", USD_to_JPY(amount), "JPY")
    elif choice == 4:
        print(amount, "USD =", USD_to_CAD(amount), "CAD")

    displayMenu()
    choice = int(input())
```

2. Solution

```
def displayMenu():
    print("-----------------------------------------------")
    print("1. Convert USD to Euro (EUR)")
```

```
    print("2. Convert USD to British Pound Sterling (GBP)")
    print("3. Convert EUR to USD")
    print("4. Convert EUR to GBP")
    print("5. Convert GBP to USD")
    print("6. Convert GBP to EUR")
    print("7. Exit")
    print("-----------------------------------------------")
    print("Enter a choice: ", end = "")

def USD_to_EUR(value):
    return value * 0.94

def USD_to_GBP(value):
    return value * 0.81

#Main code starts here
displayMenu()
choice = int(input())
while choice != 7:
    amount = float(input("Enter an amount: "))
    if choice == 1:
        print(amount, "USD =", USD_to_EUR(amount), "Euro")
    elif choice == 2:
        print(amount, "USD =", USD_to_GBP(amount), "GBP")
    elif choice == 3:
        print(amount, "EUR =", 1 / USD_to_EUR(1 / amount), "USD")
    elif choice == 4:
        print(amount, "EUR =", USD_to_GBP(1 / USD_to_EUR(1 / amount)), "GBP")
    elif choice == 5:
        print(amount, "GBP =", 1 / USD_to_GBP(1 / amount), "USD")
    elif choice == 6:
        print(amount, "GBP =", USD_to_EUR(1 / USD_to_GBP(1 / amount)), "EUR")

    displayMenu()
    choice = int(input())
```

3. Solution

```
from math import pi
ACCURACY = 0.000000001

def factorial(n):
    returnValue = 1
    for i in range(1, n + 1):
        returnValue *= i
    return returnValue

def mySin(x):
    sign = 1
```

```
    sinus = 0
    i = 1
    while True:
        sinusPrevious = sinus
        sinus += sign * x ** i / factorial(i)

        sign = -sign
        i += 2
        if abs(sinus - sinusPrevious) <= ACCURACY: break
    return sinus

def degreesToRad(degrees):
    return 2 * pi * degrees / 360

#Main code starts here
for i in range(361):
    print("sin(", i, ") ~= ", mySin(degreesToRad(i)), sep = "")
```

4. Solution

```
def isLeap(year):
    returnValue = False
    if year % 4 == 0 and year % 100 != 0 or year % 400 == 0:
        returnValue = True
    return returnValue

def numOfDays(year, month):
    if month in [4, 6, 9, 11]:
        days = 30
    elif month == 2:
        if isLeap(year):
            days = 29
        else:
            days = 28
    else:
        days = 31

    return days

def checkDate(day, month, year):
    returnValue = True
    if month not in range(1, 13):
        returnValue = False
    elif day < 1 or day > numOfDays(year, month):
        returnValue = False
    return returnValue

#Main code starts here
day = int(input("Enter day: "))
```

```
month = int(input("Enter month: "))
year = int(input("Enter year: "))
while not checkDate(day, month, year):
    print("Error!")
    day = int(input("Enter day: "))
    month = int(input("Enter month: "))
    year = int(input("Enter year: "))

total = 0
for i in range(1, month):
    total += numOfDays(year, i)
total += day

print(total)
```

5. Solution

```
from random import randrange

def dice():
    return randrange(1, 7)

#Main code starts here
names = [None] * 2
names[0] = input("Player1 - Enter name: ")
names[1] = input("Player2 - Enter name: ")

for player in range(2):
    total = 0
    for i in range(10):
        print("Player " + names[player] + ", hit enter to roll the dice!")
        input()  #This statement just waits the user to hit the enter key

        dice1 = dice()
        dice2 = dice()
        print(dice1, dice2)
        total += dice1 + dice2
    if player == 0:
        totalPlayer1 = total
    else:
        totalPlayer2 = total

if totalPlayer1 == totalPlayer2:
    print("Tie!")
elif totalPlayer1 > totalPlayer2:
    print(names[0], " wins")
else:
    print(names[1], " wins")
```

6. Solution

```
GAS = 1
DIESEL = 2
HYBRID = 3
TAX_RATE = 0.1
CARS = 40

def getChoice():
    print("1. Gas")
    print("2. Diesel")
    print("3. Hybrid")
    return int(input("Enter type of the car: "))

def getDays():
    return int(input("Enter total number of rental days: "))

def getCharge(carType, rentalDays):
    if carType == GAS:
        if rentalDays <= 5:
            charge = rentalDays * 24
        elif rentalDays <= 8:
            charge = 5 * 24 + (rentalDays - 5) * 22
        else:
            charge = 5 * 24 + 3 * 22 + (rentalDays - 8) * 18
    elif carType == DIESEL:
        if rentalDays <= 5:
            charge = rentalDays * 28
        elif rentalDays <= 8:
            charge = 5 * 28 + (rentalDays - 5) * 25
        else:
            charge = 5 * 28 + 3 * 25 + (rentalDays - 8) * 21
    else:
        if rentalDays <= 5:
            charge = rentalDays * 30
        elif rentalDays <= 8:
            charge = 5 * 30 + (rentalDays - 5) * 28
        else:
            charge = 5 * 30 + 3 * 28 + (rentalDays - 8) * 23
    charge = charge * (1 + TAX_RATE) #This is equivalent to charge += charge * TAX_RATE
    return charge

#Main code starts here
rentedCarTypes = [None] * CARS
rentedDays = [None] * CARS

for i in range(CARS):
    rentedCarTypes[i] = getChoice()
    rentedDays[i] = getDays()
```

```
total = 0
for i in range(CARS):
    charge = getCharge(rentedCarTypes[i], rentedDays[i])
    print("Car No", (i + 1), ":", charge)
    total += charge

count = 0
for i in range(CARS):
    if rentedCarTypes[i] == HYBRID:
        count += 1

print("Hybrids rented:", count)
print("Net profit:", total / (1 + TAX_RATE))
```

7. Solution

```
CHANNELS = 10
DAYS = 7

def getData(names, viewers):
    dayNames = ["Monday", "Tuesday", "Wednesday", "Thursday", "Friday", "Saturday", "Sunday"]

    for i in range(CHANNELS):
        names[i] = input("Enter name for channel No." + str(i + 1) + ": ")
        for j in range(DAYS):
            viewers[i][j] = int(input("Enter the number of viewers of the main news program on " +   \
                                      dayNames[j] + " for channel " + names[i] + ": "))

def getAverage(a):
    total = 0           # Or you can do the following
    for i in range(5):  #
        total += a[i]   #
    return total / 5    # return fsum(total) / 5

#Main code starts here
names = [None] * CHANNELS
viewers = [[None] * DAYS for i in range(CHANNELS)]
getData(names, viewers)

for i in range(CHANNELS):
    weekend = (viewers[i][DAYS - 2] + viewers[i][DAYS - 1]) / 2
    if weekend >= 1.2 * getAverage(viewers[i]):    # viewers[i] represents the whole row
        print(names[i])

for i in range(CHANNELS):
    increasing = True
    for j in range(1, DAYS):
        if viewers[i][j] <= viewers[i][j - 1]:
```

```
            increasing = False
    if increasing:
        print(names[i])
```

8. Solution

```
CITIZENS = 300

def inputData(SSNs, answers):
    for i in range(CITIZENS):
        SSNs[i] = int(input("Enter SSN: "))
        answers[i] = input("Enter answer: ")

def sortLists(SSNs, answers):
    for m in range(CITIZENS):
        minimum = SSNs[m]
        indexOfMin = m
        for n in range(m, CITIZENS):
            if SSNs[n] < minimum:
                minimum = SSNs[n]
                indexOfMin = n
        SSNs[m], SSNs[indexOfMin] = SSNs[indexOfMin], SSNs[m]
        answers[m], answers[indexOfMin] = answers[indexOfMin], answers[m]

def searchList(SSNs, SSN):
    left = 0
    right = CITIZENS - 1
    found = False
    while left <= right and not found:
        middle = (left + right) // 2

        if SSN < SSNs[middle]:
            right = middle - 1
        elif SSN > SSNs[middle]:
            left = middle + 1
        else:
            found = True

    if not found:
        print("SSN not found!")
        return -1
    else:
        return middle

def countAnswers(answers, answer):

    count = 0
    for i in range(CITIZENS):
        if answers[i] == answer:
```

```
            count += 1
    return count

#Main code starts here
SSNs = [None] * CITIZENS
answers = [None] * CITIZENS

while True:
    inputData(SSNs, answers)
    sortLists(SSNs, answers)

    SSN = int(input("Enter an SSN to search: "))

    index = searchList(SSNs, SSN)
    if index != -1:
        answer = answers[index]
        print(answer)

        count = countAnswers(answers, answer)
        print(count * 100 / CITIZENS)
    answer = input("Repeat? ")
    if answer != "yes": break
```

9. Solution

```
TEAMS = 8
GAMES = 12

def inputData(names, results):
    for i in range(TEAMS):
        names[i] = input("Enter team name: ")
        for j in range(GAMES):
            results[i][j] = input("Enter result (W, L, T): ")

def displayResult(names, results):
    result = input("Enter a result to search (W, L, T): ")
    for i in range(TEAMS):
        print("Team:", names[i])
        found = False
        for j in range(GAMES):
            if results[i][j] == result:
                print("Week:", (j + 1))
                found = True
        if not found:
            print("nothing found")

def findTeam(names):
    name = input("Enter a name to search: ")
```

```
    i = 0
    while i < TEAMS - 1 and names[i] != name:
        i += 1

    if names[i] != name:
        returnValue = -1
    else:
        returnValue = i
    return returnValue

#Main code starts here
names = [None] * TEAMS
results = [[None] * GAMES for i in range(TEAMS)]

inputData(names, results)
displayResult(names, results)

index = findTeam(names)
while index != -1:
    total = 0
    for j in range(GAMES):
        if results[index][j] == "W":
            total += 3
        elif results[index][j] == "T":
            total += 1
    print("Points:", total)
    index = findTeam(names)

if index == -1:
    print "Team not found"
```

10. Solution

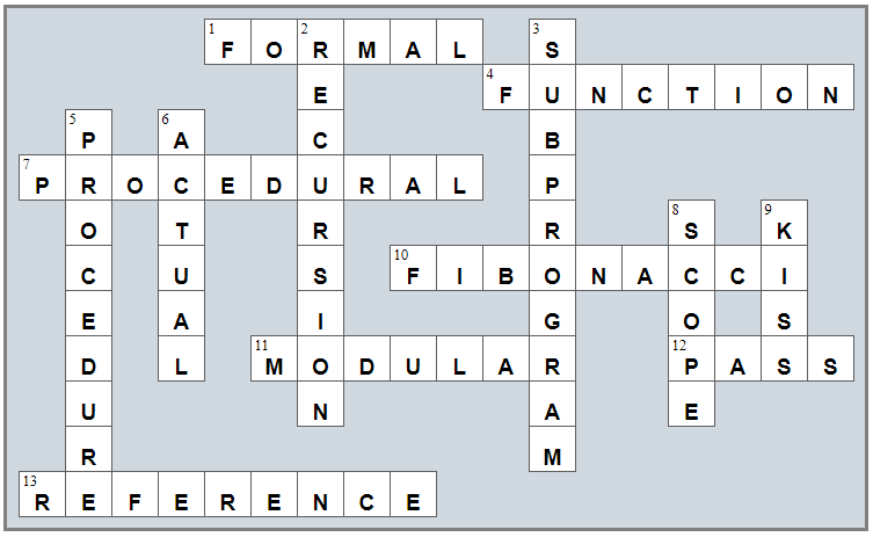
```
def hasDuplicateDigits(num):
    #Initialize an array to store the count of each digit
    digitCount = [0] * 10

    while num > 0:
        digit = num % 10   #Extract the last digit
        if digitCount[digit] > 0:
            return True #If this digit has been seen before, return true

        digitCount[digit] += 1   #Increment the count of this digit
        num //= 10   #Move to the next digit

    return False  #No duplicate digits found

num = int(input("Enter an integer: "))
while num < 11:
    num = int(input("Wrong number! Enter an integer greater than 10: "))
```

```
if hasDuplicateDigits(num):
    print("The integer contains duplicate digits")
else:
    print("The integer does not contain duplicate digits")
```

Review in “Subprograms”

Review Crossword Puzzle

1.



Chapter 38

38.9 Review Questions: True/False

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

38.10 Review Exercises

1. Solution

```
class Geometry:
    def rectangleArea(self, b, h):
        return b * h

    def triangleArea(self, b, h):
        return b * h / 2

#Main code starts here
gmtr = Geometry()

sqrSide = float(input("Enter square side: "))

rctnglBase = float(input("Enter rectangle base: "))
rctnglHeight = float(input("Enter rectangle height: "))

trnglBase = float(input("Enter triangle base: "))
trnglHeight = float(input("Enter triangle height: "))

print(gmtr.rectangleArea(sqrSide, sqrSide))
print(gmtr.rectangleArea(rctnglBase, rctnglHeight))
print(gmtr.triangleArea(trnglBase, trnglHeight))
```

2. Solution

```
class Pet:
    def __init__(self):
        self.kind = None
        self.legsNumber = None

    def startRunning(self):
        print("Pet is running")

    def stopRunning(self):
        print("Pet stopped")
```

```
#Main code starts here
pet1 = Pet()
pet1.kind = "dog"
pet1.legsNumber = 4

pet2 = Pet()
pet2.kind = "monkey"
pet2.legsNumber = 2

pet1.startRunning()
pet2.startRunning()
pet1.stopRunning()
```

3. Solution

```
class Pet:
    def __init__(self, kind, legsNumber):
        #Initialize fields (using the corresponding properties)
        self.kind = kind
        self.legsNumber = legsNumber

    #Define the getter
    @property
    def kind(self):
        return self._kind

    #Define the setter
    @kind.setter
    def kind(self, value):
        if value != "":
            self._kind = value
        else:
            raise ValueError("Cannot be empty")

    #Define the getter
    @property
    def legsNumber(self):
        return self._legsNumber

    #Define the setter
    @legsNumber.setter
    def legsNumber(self, value):
        if value >= 0:
            self._legsNumber = value
        else:
            raise ValueError("Cannot be negative")

    def startRunning(self):
```

```
        print("Pet is running")

    def stopRunning(self):
        print("Pet stopped")

#Main code starts here
pet1 = Pet("dog", 4)

pet1.startRunning()
pet1.stopRunning()

pet1.kind = ""          #This will throw an error
pet1.legsNumber = -3   #This will throw an error
```

4. Solution

```
BOXES = 30

class Box:
    def __init__(self, width, length, height):
        self._width = width
        self._length = length
        self._height = height

    def displayVolume(self):
        print("Volume", self._width * self._length * self._height)

    def displayDimensions(self):
        print(self._width, "x", self._length, "x", self._height)

#Main code starts here
listOfObj = [None] * BOXES  #create a list

for i in range(BOXES):
    w = float(input("Enter width: "))
    l = float(input("Enter length: "))
    h = float(input("Enter height: "))

    #add each new object to the list
    listOfObj[i] = Box(w, l, h)

for i in range(BOXES):
    listOfObj[i].displayDimensions()
    listOfObj[i].displayVolume()
```

5. Solution

```
BOXES = 30

class Box:
```

```
    def __init__(self, width, length, height):
        #Initialize fields (using the corresponding properties)
        self.width = width
        self.length = length
        self.height = height

    #Define the getter
    @property
    def width(self):
        return self._width

    #Define the setter
    @width.setter
    def width(self, value):
        if value > 0:
            self._width = value
        else:
            raise ValueError("Cannot be negative or zero")

    #Define the getter
    @property
    def length(self):
        return self._length

    #Define the setter
    @length.setter
    def length(self, value):
        if value > 0:
            self._length = value
        else:
            raise ValueError("Cannot be negative or zero")

    #Define the getter
    @property
    def height(self):
        return self._height

    #Define the setter
    @height.setter
    def height(self, value):
        if value > 0:
            self._height = value
        else:
            raise ValueError("Cannot be negative or zero")

    def displayVolume(self):
        print("Volume", self.width * self.length * self.height)

    def displayDimensions(self):
```

```
        print(self.width, "x", self.length, "x", self.height)

#Main code starts here
listOfObj = [None] * BOXES   #create a list

for i in range(BOXES):
    w = float(input("Enter width: "))
    l = float(input("Enter length: "))
    h = float(input("Enter height: "))

    #add each new object to the list
    listOfObj[i] = Box(w, l, h)

for i in range(BOXES):
    listOfObj[i].displayDimensions()
    listOfObj[i].displayVolume()
```

6. Solution

```
class Cube:
    def __init__(self, edge):
        self._edge = edge

    def displayVolume(self):
        print("Volume:", self._edge ** 3)

    def displayOneSurface(self):
        print("One surface:", self._edge ** 2)

    def displayTotalSurface(self):
        print("Total surface:", 6 * self._edge ** 2)

#Main code starts here
edge = float(input("Enter edge length of a cube: "))

cube1 = Cube(edge)

cube1.displayVolume()
cube1.displayOneSurface()
cube1.displayTotalSurface()
```

7. Solution

```
class Cube:
    def __init__(self, edge):
        #Initialize field (using the corresponding property)
        self.edge = edge

    #Define the getter
    @property
```

```
    def edge(self):
        return self._edge

    #Define the setter
    @edge.setter
    def edge(self, value):
        if value > 0:
            self._edge = value
        else:
            raise ValueError("Cannot be negative or zero")

    def displayVolume(self):
        print("Volume:", self.edge ** 3)

    def displayOneSurface(self):
        print("One surface:", self.edge ** 2)

    def displayTotalSurface(self):
        print("Total surface:", 6 * self.edge ** 2)

#Main code starts here
edge = float(input("Enter edge length of a cube: "))

cube1 = Cube(edge)

cube1.displayVolume()
cube1.displayOneSurface()
cube1.displayTotalSurface()
```

8. Solution

```
class Circle:
    def __init__(self):
        self._radius = -1   #Private field

    #Define the getter
    @property
    def radius(self):
        if self._radius != -1:
            return self._radius
        else:
            raise ValueError("Radius is not set")

    #Define the setter
    @radius.setter
    def radius(self, value):
        if value > 0:
            self._radius = value
        else:
```

```
            raise ValueError("Cannot be negative or zero")

    def getDiameter(self):
        return 2 * self.radius

    def getArea(self):
        return 3.14 * self.radius ** 2

    def getPerimeter(self):
        return 2 * 3.14 * self.radius

def displayMenu():
    print("1. Enter radius")
    print("2. Display radius")
    print("3. Display diameter")
    print("4. Display area")
    print("5. Display perimeter")
    print("6. Exit")

#Main code starts here
circle1 = Circle()

while True:
    displayMenu()

    choice = int(input("Enter a choice: "))

    if choice == 1:
        radius = float(input("Enter radius: "))
        circle1.radius = radius
    elif choice == 2:
        print("Radius:", circle1.radius)
    elif choice == 3:
        print("Diameter:", circle1.getDiameter())
    elif choice == 4:
        print("Area:", circle1.getArea())
    elif choice == 5:
        print("Perimeter:", circle1.getPerimeter())
    elif choice == 6:
        break
```

9. Solution

```
class Info:
    #Define the getter
    @property
    def userText(self):
        return self._userText
```

```
    #Define the setter
    @userText.setter
    def userText(self, value):
        if value != "":
            self._userText = value
        else:
            raise ValueError("Cannot be set to empty")

    def getSpacesCount(self):
        count = 0
        for char in self.userText:
            if char == " ":
                count += 1
        return count

    def getWordsCount(self):
        return self.getSpacesCount() + 1

    def getVowelsCount(self):
        count = 0
        for char in self.userText.lower():
            if char in "aeiou":
                count += 1
        return count

    def getLettersCount(self):
        return len(self.userText) - self.getSpacesCount()

#Main code starts here
inf = Info()

text = input("Enter a text: ")

inf.userText = text

print("Text:", inf.userText)
print("Spaces:", inf.getSpacesCount())
print("Words:", inf.getWordsCount())
print("Vowels:", inf.getVowelsCount())
print("Total number of letters:", inf.getLettersCount())
```

10. Solution

```
class EncryptDecrypt:
    def __init__(self):
        self._encrDecrKey = -1  #Private field. #It does not call the setter!
        self._alphabet = " abcdefghijklmnopqrstuvwxyz"  #Space is a valid character!

    #Define the getter
```

```
    @property
    def encrDecrKey(self):
        if self._encrDecrKey != -1:
            return self._encrDecrKey
        else:
            raise ValueError("Key is not set")

    #Define the setter
    @encrDecrKey.setter
    def encrDecrKey(self, value):
        if value in range(1,27):
            self._encrDecrKey = value
        else:
            raise ValueError("Key must be between 1 and 26")

    def encrypt(self, message):
        returnValue = ""
        for char in message:
            index = self._alphabet.find(char)
            newIndex = index + self.encrDecrKey
            if newIndex >= 27:
                newIndex -= 27
            newLetter = self._alphabet[newIndex]
            returnValue += newLetter
        return returnValue

    def decrypt(self, encMessage):
        returnValue = ""
        for char in encMessage:
            index = self._alphabet.find(char)
            newIndex = index - self.encrDecrKey
            if newIndex < 0:
                newIndex += 27
            newLetter = self._alphabet[newIndex]
            returnValue += newLetter
        return returnValue

def displayMenu():
    print("1. Enter encryption/decryption key")
    print("2. Encrypt a message")
    print("3. Decrypt a message")
    print("4. Exit")

#Main code starts here
ed = EncryptDecrypt()

displayMenu()
choice = int(input("Enter a choice: "))
while choice != 4:
```

```
    if choice == 1:
        encrDecrKey = int(input("Enter encryption/decryption key: "))
        ed.encrDecrKey = encrDecrKey
    elif choice == 2:
        text = input("Enter message to encrypt: ")
        print("Encrypted message:", ed.encrypt(text))
    else:
        text = input("Enter message to decrypt: ")
        print("Decrypted message:", ed.decrypt(text))

    displayMenu()
    choice = int(input("Enter a choice: "))
```

11. Solution

```
class Vehicle:
    #Define the constructor
    def __init__(self, numberOfWheels, color, length, width, height):
        self.numberOfWheels = numberOfWheels
        self.color = color
        self.length = length
        self.width = width
        self.height = height

    def startEngine(self):
        print("The engine started")

    def stopEngine(self):
        print("The engine stopped")

class Car(Vehicle):
    #Define the constructor
    def __init__(self, numberOfWheels, color, length, width, height):
        super().__init__(numberOfWheels, color, length, width, height)
        self.bootCapacity = 0

    def turnWindshieldWipersOn(self):
        print("The windshield wipers have been turned on!")

class Motorcycle(Vehicle):
    #Define the constructor
    def __init__(self, numberOfWheels, color, length, width, height):
        super().__init__(numberOfWheels, color, length, width, height)
        self.hasLuggage = False

    def doAWheelie(self):
        print("I am doing a wheelie!!!")

#Main code starts here
car1 = Car(4, "Red", 5, 2, 1.5)
```

```
car1.bootCapacity = 300
car1.startEngine()
car1.turnWindshieldWipersOn()
car1.stopEngine()

car2 = Car(4, "Green", 4.5, 2.2, 1.4)
car2.bootCapacity = 400
car2.startEngine()
car2.turnWindshieldWipersOn()
car2.stopEngine()

motorcycle1 = Motorcycle(2, "Blue", 2, 0.9, 1.3)
motorcycle1.hasLuggage = True
motorcycle1.startEngine()
motorcycle1.doAWheelie()
motorcycle1.stopEngine()
```

12. Solution

```
class SchoolMember:
    def __init__(self, name, age):
        self.setName(name)
        self.setAge(age)
        print("A school member was initialized")

    def getName(self):
        return self._name

    def setName(self, value):
        if value != "":
            self._name = value
        else:
            raise ValueError("Name cannot be empty")

    def getAge(self):
        return self._age

    def setAge(self, value):
        if value > 0:
            self._age = value
        else:
            raise ValueError("Age cannot be negative or zero")

class Teacher(SchoolMember):
    def __init__(self, name, age, salary):
        super().__init__(name, age)

        self.setSalary(salary)

        print("A teacher was initialized")

    def getSalary(self):
```

```
        return self._salary

    def setSalary(self, value):
        if value >= 0:
            self._salary = value
        else:
            raise ValueError("Salary cannot be negative")

    def displayValues(self):
        print("Name:", self.getName())
        print("Age:", self.getAge())
        print("Salary:", self.getSalary())

class Student(SchoolMember):
    def __init__(self, name, age, finalGrade):
        super().__init__(name, age)

        self.setFinalGrade(finalGrade)

        print("A student was initialized")

    def getFinalGrade(self):
        return self._finalGrade

    def setFinalGrade(self, value):
        if value in ["A", "B", "C", "D", "E", "F"]:
            self._finalGrade = value
        else:
            raise ValueError("Final grade must be in the range of 'A' to 'F'")

    def displayValues(self):
        print("Name:", self.getName())
        print("Age:", self.getAge())
        print("Final grade:", self.getFinalGrade())

#Main code starts here
teacher1 = Teacher("Mr. John Scott", 43, 35000)
teacher2 = Teacher("Mrs. Ann Carter", 5, 32000)

student1 = Student("Peter Nelson", 14, "A")
student2 = Student("Helen Morgan", 13, "B")

teacher1.displayValues()
teacher2.displayValues()
student1.displayValues()
student2.displayValues()
```

13. Solution

```
class SchoolMember:
    def __init__(self, name, age):
        self.name = name
```

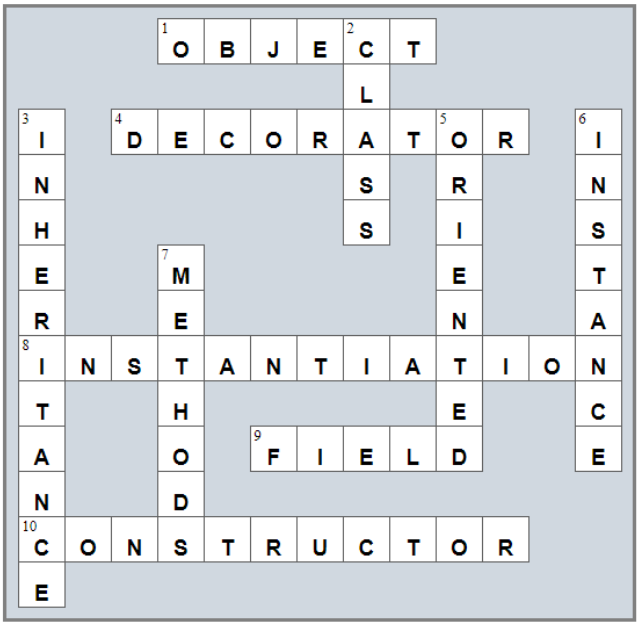
```
        self.age = age
        print("A school member was initialized")

    @property
    def name(self):
        return self._name

    @name.setter
    def name(self, value):
        if value != "":
            self._name = value
        else:
            raise ValueError("Name cannot be empty")

    @property
    def age(self):
        return self._age

    @age.setter
    def age(self, value):
        if value > 0:
            self._age = value
        else:
            raise ValueError("Age cannot be negative or zero")

class Teacher(SchoolMember):
    def __init__(self, name, age, salary):
        super().__init__(name, age)
        self.salary = salary
        print("A teacher was initialized")

    def displayValues(self):
        print("Name:", self.name)
        print("Age:", self.age)
        print("Salary:", self.salary)

    @property
    def salary(self):
        return self._salary

    @salary.setter
    def salary(self, value):
        if value >= 0:
            self._salary = value
        else:
            raise ValueError("Salary cannot be negative")

class Student(SchoolMember):
    def __init__(self, name, age, finalGrade):
        super().__init__(name, age)
        self.finalGrade = finalGrade
```

```
        print("A student was initialized")

    def displayValues(self):
        print("Name:", self.name)
        print("Age:", self.age)
        print("Final grade:", self.finalGrade)

    @property
    def finalGrade(self):
        return self._finalGrade

    @finalGrade.setter
    def finalGrade(self, value):
        if value in ["A", "B", "C", "D", "E", "F"]:
            self._finalGrade = value
        else:
            raise ValueError("Final grade must be in the range of 'A' to 'F'")

#Main code starts here
teacher1 = Teacher("Mr. John Scott", 43, 35000)
teacher2 = Teacher("Mrs. Ann Carter", 5, 32000)

student1 = Student("Peter Nelson", 14, "A")
student2 = Student("Helen Morgan", 13, "B")

teacher1.displayValues()
teacher2.displayValues()
student1.displayValues()
student2.displayValues()
```

Review in “Object Oriented Programming”

Review Crossword Puzzle

1.



Chapter 39

39.8 Review Questions: True/False

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

39.9 Review Exercises

1. Solution

```
PATH = "c:/temp/"

days = ["Sunday", "Monday", "Tuesday", "Wednesday", "Thursday", "Friday", "Saturday"]

f = open(PATH + "days_of_week.txt", "w")
for d in days:
    f.write(d + "\n")
f.close()
```

2. Solution

```
PATH = "c:/temp/"

days = []

f = open(path + "days_of_week.txt", "r")
for line in f:
    days.append(line)
f.close()

for day in days[::-1]:
    print(day, end = "")
```

3. Solution

```
PATH = "c:/temp/"

f = open(PATH + "days_of_week.txt", "a")
f.write("*** End of File ***")
f.close()
```

4. Solution

```
PATH = "c:/temp/"
```

```
from random import randrange

f = open(PATH + "randoms.txt", "w")
for i in range(50):
    f.write(str(randrange(1, 101)) + "\n")
f.close()
```

5. Solution

```
PATH = "c:/temp/"

from random import randrange

for i in range(1, 11):
    f = open(PATH + "file" + str(i) + ".txt", "w")
    f.write(str(randrange(100, 1000)))
    f.close()
```

6. Solution

```
PATH = "c:/temp/"

f = open(PATH + "multiplication_table.txt", "w")

for i in range(1, 11):
    for j in range(1, 5):
        f.write(str(i) + " x " + str(j) + " = " + str(i * j) + "\n")

f.close()
```

7. Solution

```
PATH = "c:/temp/"

f = open(PATH + "a_file.txt", "r")

for line in f:
    print(len(line) - 1)  #Minus 1 due to \n at the end of the line

f.close()
```

8. Solution

```
PATH = "c:/temp/"

f = open(PATH + "a_file.txt", "r")

countLines = 0
countChars = 0
for line in f:
    countLines += 1
    countChars += len(line) - 1
```

```
f.close()

print("Total characters: ", countChars)
print("Total Lines: ", countLines)
```

9. Solution

First approach

```
PATH = "c:/temp/"

f = open(PATH + "a_file.txt", "r")

i = 1
for line in f:
    for character in line:
        if character in ",.!":
            print("There is a punctuation mark on line No", i)
            break
    i += 1

f.close()
```

Second approach

```
PATH = "c:/temp/"

f = open(PATH + "a_file.txt", "r")

i = 1
for line in f:
    if "," in line or "." in line or "!" in line:
        print("There is a punctuation mark on line No", i)

    i += 1

f.close()
```

Chapter 40

40.2 Review Exercises

1. Solution

First approach

```
PATH = "c:/temp/"

fin = open(PATH + "f_data40.2-1.txt")
values = fin.readline()
fin.close()

total = 0
count = 0
for i in range(10):
    number = int(values[i * 3 : i * 3 + 2])
    if number > 50:
        total += number
        count += 1

if count > 0:
    print(total / count)
```

Second approach

```
PATH = "c:/temp/"

fin = open(PATH + "f_data40.2-1.txt")

total = 0
count = 0
for i in range(10):
    number = int(fin.read(2))
    if number > 50:
        total += number
        count += 1
    space = fin.read(1)

fin.close()

if count > 0:
    print(total / count)
```

2. Solution

First approach

```
PATH = "c:/temp/"

fin = open(PATH + "f_data40.2-2.txt")
values = fin.readline()
fin.close()
```

```
total = 0
count = 0
i = 0
while i < len(values) // 4:
    number = int(values[i * 4 : i * 4 + 3])
    if 300 <= number <= 500:
        total += number
        count += 1
    i += 1

if count > 0:
    print(total / count)
```

Second approach

```
PATH = "c:/temp/"

fin = open(PATH + "f_data40.2-2.txt")
total = 0
count = 0

while True:
    number = int(fin.read(3))
    if 300 <= number <= 500:
        total += number
        count += 1
    comma = fin.read(1)
    if comma == "": break

fin.close()

if count > 0:
    print(total / count)
```

3. Solution

```
PATH = "c:/temp/"

fin = open(PATH + "f_data40.2-3.txt")
#Read the first line
line = fin.readline()

commaPosition = line.find(",")
grade = int(line[:commaPosition])
name = line[commaPosition + 1:].strip()

maximum = minimum = grade
maxName = minName = name

#Read the rest of the lines
for line in fin:
    commaPosition = line.find(",")
```

```
    grade = int(line[:commaPosition])
    name = line[commaPosition + 1:].strip()

    if grade > maximum:
        maximum = grade
        maxName = name

    if grade < minimum:
        minimum = grade
        minName = name

fin.close()

print(maxName)
print(minName)
```

4. Solution

```
PATH = "c:/temp/"

keyword = input("Enter keyword to search: ").lower()

fin = open(PATH + "f_data40.2-4.txt")

maximum = total = 0
stringInfo1 = stringInfo2 = ""
for line in fin:
    width = float(line[0:5])
    length = float(line[6:11])
    height = float(line[12:17])
    description = line[18:].strip()
    if description.lower().find(keyword) != -1:
        stringInfo1 += description + " - Dimensions: " +
                       str(width) + " x " + str(length) + " x " + str(height) + "\n"

    volume = width * length * height / 1728
    stringInfo2 += description + " - Volume = " + str(volume) + " cubic feet\n"

    total += volume

    if volume > maximum:
        maximum = volume
        maximumDescription = description

fin.close()

if stringInfo1 != "":
    print("Keyword '" + keyword + "' found!")
    print(stringInfo1)

print("Volume of each item:")
print(stringInfo2)
```

```
print("Total volume:", total)
print("Greatest box:", maximumDescription)
```

5. Solution

First approach

```
filename1 = input("Enter filename No 1: ")

if filename1[-4:] != ".txt":
    print("Wrong filename")
else:
    filename2 = input("Enter filename No 2: ")
    if filename2[-4:] != ".txt":
        print("Wrong filename")
    else:
        fin = open(filename2)
        content = fin.read()
        fin.close()

        fin = open(filename1)
        content += fin.read()  #Concatenation
        fin.close()

        fout = open("final.txt", "w")
        fout.write(content)
        fout.close()
```

Second approach

```
filename1 = input("Enter filename No 1: ")

if filename1[-4:] != ".txt":
    print("Wrong filename")
else:
    filename2 = input("Enter filename No 2: ")
    if filename2[-4:] != ".txt":
        print("Wrong filename")
    else:
        fin1 = open(filename1)
        fin2 = open(filename2)
        fout = open("final.txt", "w")

        fout.write(fin2.read() + fin1.read())

        fout.close()
        fin2.close()
        fin1.close()
```

6. Solution

```
PATH = "c:/temp/"

fin = open(PATH + "f_data40.2-6.txt")
```

```
numbers = []
for line in fin:
    numbers.append(float(line))

fin.close()

elements = len(numbers)  #Get the size of the list

#Bubble sort
for m in range(elements - 1):
    for n in range(elements - 1, m, -1):
        if numbers[n] < numbers[n - 1]:
            numbers[n], numbers[n - 1] = numbers[n - 1], numbers[n]

fout = open(PATH + "f_data40.2-6.txt", "a")
fout.write("********* Sorted numbers *************\n")
for number in numbers:
    fout.write(str(number) + "\n")

fout.close()
```

7. Solution

```
PATH = "c:/temp/"
NUMBER_OF_CITIES = 8

fin = open(PATH + "f_data40.2-7.txt", "r")

cities = []
temperatures = []
onCityLine = True
for line in fin:
    if onCityLine:
        cities.append(line)
    else:
        temperatures.append(float(line))

    onCityLine = not onCityLine  #Toggle between True and Talse with each iteration of the code

fin.close()

total = 0
for i in range(NUMBER_OF_CITIES):
    total += temperatures[i]

average = total / NUMBER_OF_CITIES
print(average)

maximum = max(temperatures)
print("Highest temperature:", maximum)
for i in range(NUMBER_OF_CITIES):
```

```
    if temperatures[i] == maximum:
        print(cities[i])
```

8. Solution

```
PATH = "c:/temp/"

def abbreviate(word):
    if len(word) > 10:
        return word[0] + str(len(word) - 2) + word[-1]
    else:
        return word

fin = open(PATH + "f_data40.2-8.txt", "r")

for line in fin:
    line = line.strip() #Remove \n from the end

    spaceIndex = line.find(" ")  #Find the first space
    while spaceIndex != -1:
        word = line[0:spaceIndex]     #Get the word and
        line = line[spaceIndex + 1:]  #remove the word from line

        print(abbreviate(word) + " ", end = "")

        spaceIndex = line.find(" ")

    #Display the last word remained in the string line
    print(abbreviate(line))

fin.close()
```

9. Solution

```
PATH = "c:/temp/"

def pigLatinTranslator(word):
    vowels = "aeiou"

    if word[0] in vowels: #If first character is vowel
        pigLatinWord = word + "way"
    else:
        #Find the index of the first vowel
        firstVowelIndex = -1
        for i in range(len(word)):
            if word[i] in vowels:
                firstVowelIndex = i
                break

        #If at least one vowel found
        if firstVowelIndex != -1:
            #Move the consonants to the end
```

```
            word = word[firstVowelIndex:] + word[0:firstVowelIndex]

        pigLatinWord = word + "ay"

    return pigLatinWord;

fin = open(PATH + "f_data40.2-9.txt", "r")
fout = open(PATH + "pig_latin_translation.txt", "w")

for line in fin:
    line = line.strip() #Remove \n from the end

    spaceIndex = line.find(" ")  #Find the first space
    while spaceIndex != -1:
        word = line[0:spaceIndex]     #Get the word and
        line = line[spaceIndex + 1:]  #remove the word from line

        fout.write(pigLatinTranslator(word) + " ")

        spaceIndex = line.find(" ")

    #Write the last word remained in the string line
    fout.write(pigLatinTranslator(line) + "\n")

fin.close()
fout.close()
```

10. Solution

```
PATH = "c:/temp/"
X = " ABCDEFGHIJKLMNOPQRSTUVWXYZ"  #The space character remains as is
Y = " JKWCTAMEDXSLFBYUNGRZOIQVHP"

initialMessage = input("Enter a message to encrypt: ").upper()

encryptedMessage = ""
for letter in initialMessage:
    #Search for letter in variable X
    for i in range(27):
        if letter == X[i]:
            #Create encrypted message using letters from variable Y
            encryptedMessage += Y[i]
            break

fout = open(PATH + "encrypted.txt", "w")
fout.write(encryptedMessage)
fout.close()
```

11. Solution

```
PATH = "c:/temp/"
X = " ABCDEFGHIJKLMNOPQRSTUVWXYZ"  #The space character remains as is
Y = " JKWCTAMEDXSLFBYUNGRZOIQVHP"
```

```
fin = open(PATH + "encrypted.txt")
encryptedMessage = fin.readline()
fin.close()

initialMessage = ""
for letter in encryptedMessage:
    #Search for letter in variable Y
    for i in range(27):
        if letter == Y[i]:
            #Create decrypted message using letters from variable X
            initialMessage += X[i]
            break

fout = open(PATH + "decrypted.txt", "w")
fout.write(initialMessage)
fout.close()
```

12. Solution

First approach

```
def copyFile(source, destination):
    fin = open(source, "r")
    x = fin.read()
    fin.close()

    fout = open(destination, "w")
    fout.write(x)
    fout.close()
```

Second approach

```
def copyFile(source, destination):
    fin = open(source, "r")
    fout = open(destination, "w")

    fout.write(fin.read())

    fin.close()
    fout.close()
```

13. Solution

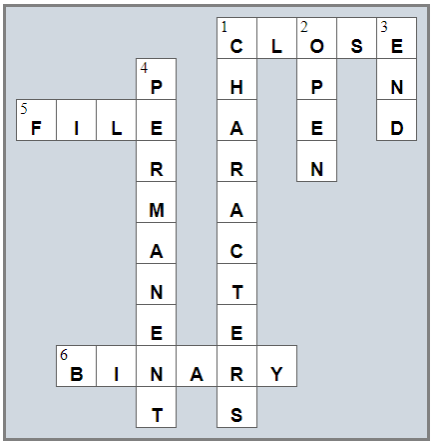
```
from math import sqrt

PATH = "c:/temp/"

class Triangle:
    #Define the constructor
    def __init__(self):
        fin = open(PATH + "f_data40.2-13.txt", "r")
        self._sideA = float(fin.readline())
        self._sideB = float(fin.readline())
```

```
        self._sideC = float(fin.readline())
        fin.close()

    def canBeTriangle(self):
        if self._sideA > 0 and self._sideB > 0 and self._sideC > 0 and   \
           self._sideA + self._sideB > self._sideC and   \
           self._sideB + self._sideC > self._sideA and   \
           self._sideC + self._sideA > self._sideB:
            return True
        else:
            return False

    def displayLengths(self):
        print("Side A:", self._sideA)
        print("Side B:", self._sideB)
        print("Side C:", self._sideC)
        if self.canBeTriangle():
            print("Can be lengths of the three sides of a triangle!")
        else:
            print("Cannot be lengths of the three sides of a triangle!")

    def displayArea(self):
        if self.canBeTriangle():
            s = (self._sideA + self._sideB + self._sideC) / 2
            area = sqrt(s * (s - self._sideA) * (s - self._sideB) * (s - self._sideC))
            print("Area:", area)

    def displayPerimeter(self):
        if self.canBeTriangle():
            perimeter = self._sideA + self._sideB + self._sideC
            print("Perimeter:", perimeter)

tr = Triangle()

tr.displayLengths()
tr.displayArea()
tr.displayPerimeter()
```

Review in “Files”

Review Crossword Puzzle

1.



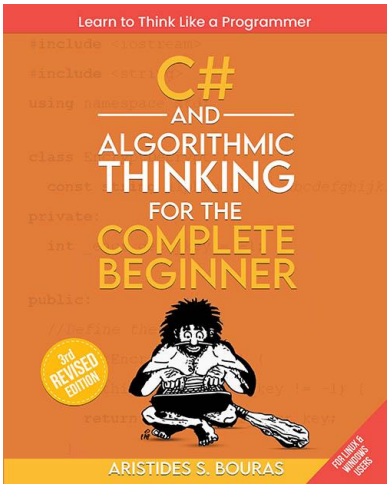
Some Final Words from the Author

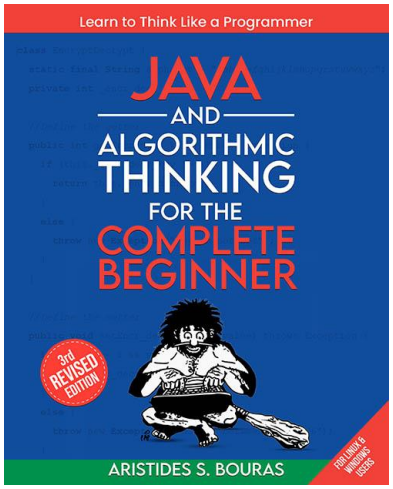
I hope you thoroughly enjoyed reading this book. I made every possible effort to ensure it is beneficial and comprehensible, even for people who may have no prior experience in programming.

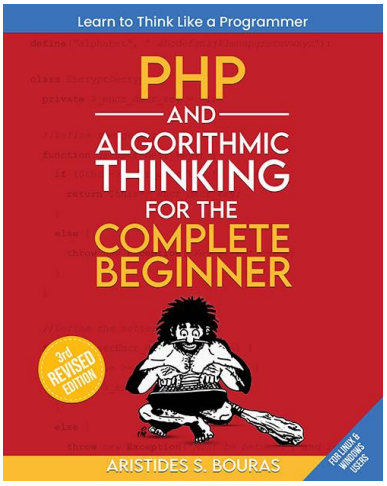
If you found this book valuable, please consider visiting the web store where you purchased it, as well as goodreads.com, to show your appreciation by writing a positive review and awarding as many stars as you think appropriate. By doing so, you will motivate me to keep writing and, of course, you'll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

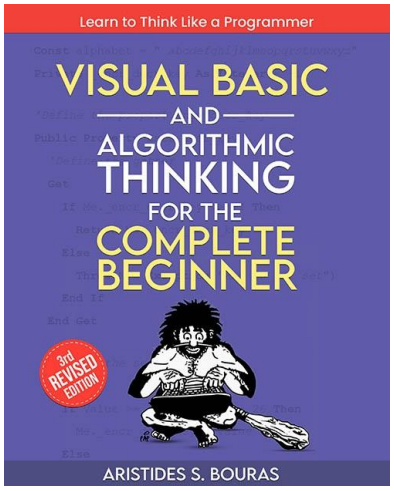
And always remember: Learning is a lifelong, continuous process that begins at birth and extends throughout your lifetime!

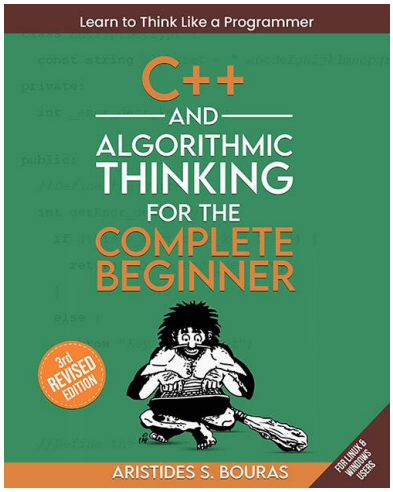
Some of my Books

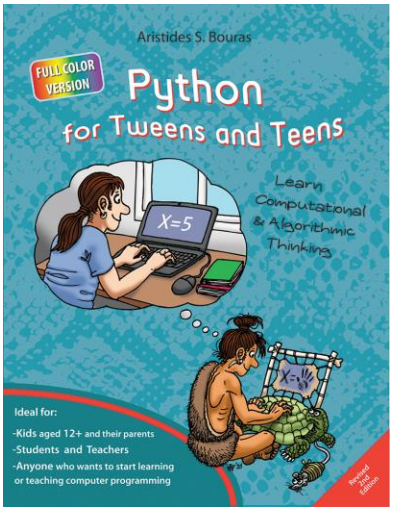


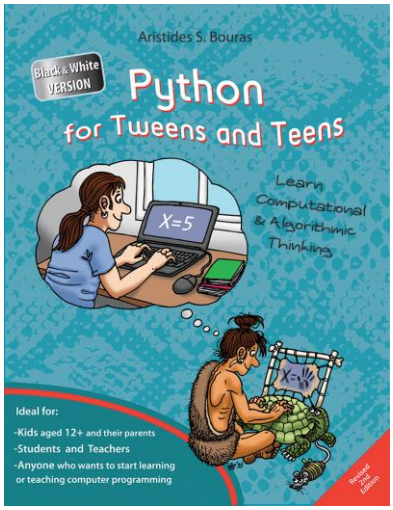


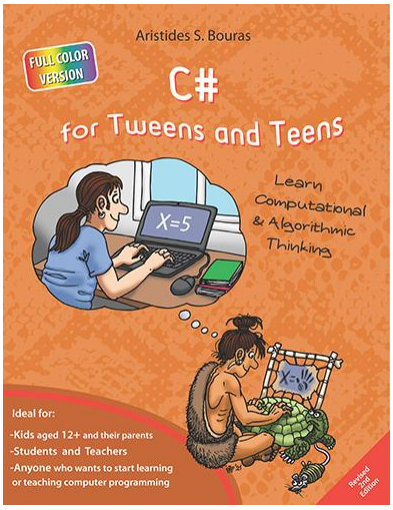


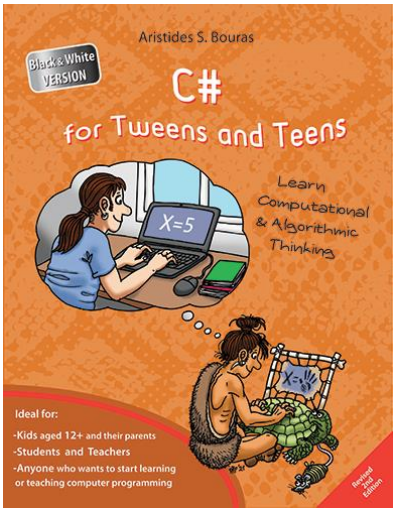


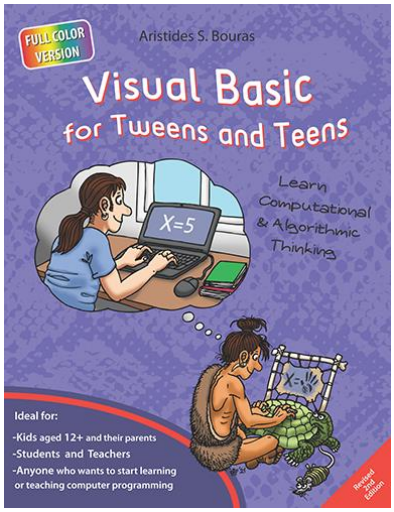


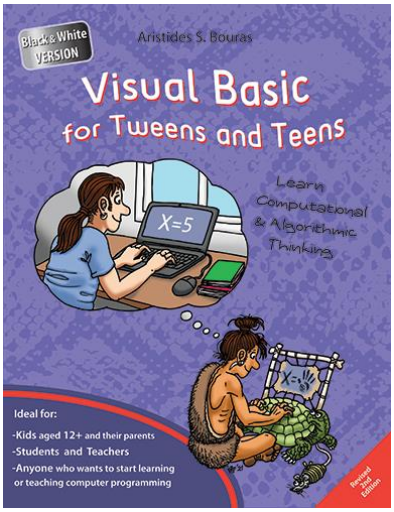












For more information about my books visit my website:

https://www.bouraspage.com