# Python for Tweens and Teens

# The Answers

By
Aristides S. Bouras
Loukia V. Ainarozidou

Python for Tweens and Teens - The Answers



http://www.bouraspage.com

RCode: 200630



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

http://www.bouraspage.com

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

# Chapter 1

## 1.7 Review Questions: True/False

1. False
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. False
21. False
22. True

## 1.8 Review Questions: Multiple Choice

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

# Chapter 3

## 3.11 Review Questions: True/False

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

## 3.12 Review Questions: Multiple Choice

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

# Chapter 4

## 4.5 Review Questions: True/False

1. False
2. True
3. False
4. True
5. False
6. True
7. False
8. True
9. False
10. True
11. True
12. True
13. False

## 4.6 Review Questions: Multiple Choice

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

## 4.7 Review Exercises

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

# Chapter 5

## 5.4 Review Questions: True/False

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

## 5.5 Review Questions: Multiple Choice

1. a
2. b
3. b

# Chapter 6

## 6.6 Review Questions: True/False

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

## 6.7 Review Questions: Multiple Choice

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

## 6.8 Review Exercises

1. ii, iv, v, ix, x
2. i – String, ii – Boolean, iii – String, iv – String, v – Float, vi – Integer
3. i – b, ii – d, iii – c, iv – e
4. i – 27, ii – 28
5. i – 0, ii – 4
6. i – 2.0, ii – 40
7. My name is Alexander the Great
8. i – 3, ii – 1
9. California California California

# Chapter 9

## 9.2 Review Exercises

### 1. Solution

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

area = base * height / 2

print(area)
```

### 2. Solution

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

k = (f + 459.67) / 1.8

print(k)
```

### 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

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

bmi = w * 703 / (h * h)

print(bmi)
```

### 7. Solution

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

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

print(days_left)
```

# Chapter 10

## 10.3 Review Questions: True/False

1. True
2. False
3. False
4. True
5. False
6. True
7. False
8. True
9. True
10. True
11. False
12. True
13. False

## 10.4 Review Exercises

1. 2
2. i – 2.5, ii – 2.2
3. i – 4, ii – 9
4. i – 5.5, ii – 3.5
5. i – 31, ii – 32

# Chapter 11

## 11.5 Review Questions: True/False

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

## 11.6 Review Questions: Multiple Choice

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

## 11.7 Review Exercises

### 1. *Solution*

```
first_name = input("First name: ")
middle_name = input("Middle name: ")
last_name = input("Last name: ")
title = input("Title: ")

print(title, first_name, middle_name, last_name)
print(first_name, middle_name, last_name)
print(last_name, ",", first_name)
print(last_name, ",", first_name, middle_name)
print(last_name, ",", first_name, middle_name, ",", title)
print(first_name, last_name)
```

### 2. *Solution*

```
import random

alphabet = "abcdefghijklmnopqrstuvwxyz"

random_word = alphabet[random.randrange(26)].upper() +    \
              alphabet[random.randrange(26)] +            \
```

```
                alphabet[random.randrange(26)] +          \
                alphabet[random.randrange(26)]

print(random_word)
```

### 3. *Solution*

```
import random
name = input("Enter name: ").lower()

#convert to lower case and remove spaces
x = name.lower().replace(" ", "")

secret_password = x[random.randrange(len(x))] +    \
                  x[random.randrange(len(x))] +    \
                  x[random.randrange(len(x))] +    \
                  str(random.randrange(1000, 10000))

print(secret_password)
```

# Chapter 12

## 12.8 Review Questions: True/False

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

## 12.9 Review Questions: Multiple Choice

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

## 12.10 Review Exercises

### 1. *Solution*

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

### 2. *Solution*

| a | b | c | a != 1 | b > a | c / 2 > 2 * a |
|---|---|---|---|---|---|
| 3 | –5 | 8 | *True* | *False* | *False* |

| | | | | | |
|---|---|---|---|---|---|
| 1 | 10 | 20 | *False* | *True* | *True* |
| –4 | –2 | –9 | *True* | *True* | *True* |

### 3. Solution

| Boolean Expression1 (BE1) | Boolean Expression2 (BE2) | BE1 or BE2 | BE1 and BE2 | not(BE2) |
|---|---|---|---|---|
| False | False | *False* | *False* | *True* |
| False | True | *True* | *False* | *False* |
| True | False | *True* | *False* | *True* |
| True | True | *True* | *True* | *False* |

### 4. Solution

| a | b | c | a > 3 or c > b and c > 1 | a > 3 and c > b or c > 1 |
|---|---|---|---|---|
| 4 | –6 | 2 | *True* | *True* |
| –3 | 2 | –4 | *False* | *False* |

### 5. Solution

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

### 6. Solution

a. `age < 12 and age != 8`

b. `6 <= age <= 9 or age == 11`

c. `age > 7 and age != 10 and age != 12`

d. `age == 6 or age == 9 or age == 11`

e. `6 <= age <= 12 and age != 8`

f. `age != 7 and age != 10`

# Chapter 13

## 13.2 Review Questions: True/False

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

## 13.3 Review Questions: Multiple Choice

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

## 13.4 Review Exercises

### 1. Solution

```
x = float(input())

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

print(x, y)
```

### 2. Solution

i. 9 12

ii. 2 2

### 3. Solution

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

### 4. Solution

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

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

### 5. Solution

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

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

### 6. Solution

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

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

## 7. Solution

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

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

## 8. 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")
```

## 9. Solution

First Approach

```
w1 = float(input("Enter the weight of the 1st person: "))
w2 = float(input("Enter the weight of the 2nd person: "))
w3 = float(input("Enter the weight of the 3rd person: "))
w4 = float(input("Enter the weight of the 4th person: "))

maximum = w1

if w2 > maximum:
    maximum = w2

if w3 > maximum:
    maximum = w3

if w4 > maximum:
    maximum = w4

print(maximum)
```

Second Approach

```
w1 = float(input("Enter the weight of the 1st person: "))
w2 = float(input("Enter the weight of the 2nd person: "))
w3 = float(input("Enter the weight of the 3rd person: "))
w4 = float(input("Enter the weight of the 4th person: "))
```

```
print(max(w1, w2, w3, w4))
```

## 10. Solution

```
a1 = int(input("Enter the age of the 1st person: "))
n1 = input("Enter the name of the 1st person: ")

a2 = int(input("Enter the age of the 2nd person: "))
n2 = input("Enter the name of the 2nd person: ")

a3 = int(input("Enter the age of the 3rd person: "))
n3 = input("Enter the name of the 3rd person: ")

a4 = int(input("Enter the age of the 4th person: "))
n4 = input("Enter the name of the 4th person: ")

minimum = a1
m_name = n1

if a2 < minimum:
    minimum = a2
    m_name = n2

if a3 < minimum:
    minimum = a3
    m_name = n3

if a4 < minimum:
    minimum = a4
    m_name = n4

print("The youngest person is", m_name)
```

## 11. Solution

First Approach

```
a1 = int(input("Enter the age of the 1st person: "))
a2 = int(input("Enter the age of the 2nd person: "))
a3 = int(input("Enter the age of the 3rd person: "))

minimum = a1
if a2 < minimum:
    minimum = a2
if a3 < minimum:
    minimum = a3
```

```
maximum = a1
if a2 > maximum:
    maximum = a2
if a3 > maximum:
    maximum = a3

middle = a1 + a2 + a3 - minimum - maximum

print(middle)
```

Second Approach

```
a1 = int(input("Enter the age of the 1st person: "))
a2 = int(input("Enter the age of the 2nd person: "))
a3 = int(input("Enter the age of the 3rd person: "))

middle = a1 + a2 + a3 - min(a1, a2, a3) - max(a1, a2, a3)

print(middle)
```

# Chapter 14

## 14.2 Review Questions: True/False

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

## 14.3 Review Questions: Multiple Choice

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

## 14.4 Review Exercises

### 1. *Solution*

i. 1

ii. 5

### 2. *Solution*

i. 7.0 18.0

ii. 0.5 3.5

### 3. *Solution*

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

### 4. Solution

```
num = float(input("Enter a number: "))
if num >= 0 and num <= 100:
    print("Given number is between 0 and 100")
else:
    print("Given number is not between 0 and 100")
```

### 5. Solution

```
num = int(input("Enter an integer: "))
if num >= 1000 and num <= 9999:
    print("Given number is a four-digit integer")
else:
    print("Given number is not a four-digit integer")
```

### 6. Solution

```
num1 = float(input("Enter first number: "))
num2 = float(input("Enter second number: "))

if num1 < num2:
    print(num1)
else:
    print(num2)
```

### 7. 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")
```

# Chapter 15

## 15.2 Review Questions: True/False

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

## 15.3 Review Exercises

### 1. Solution

i. 1 ii. 2 iii. 4 iv. 4

### 2. Solution

i. 0 5.0 ii. 10 90.0 iii. 20 160.0

### 3. Solution

First Approach

```
a = int(input("Enter an integer between -9999 and 9999: "))

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 = int(input("Enter an integer between -9999 and 9999: "))

#If variable a is negative, make it positive
if a < 0:
    a = (-1) * a

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 = "")
```

### 4. 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)")
```

### 5. 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")
```

### 6. 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")
```

### 7. Solution

```
roman = input("Enter a Roman number 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")
```

### 8. Solution

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

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

```
    print("You are awarded 10 points")
elif total == 3:
    print("You are awarded 20 points")
else:
    print("You are awarded 45 points")
```

### 9. Solution

```
num = input("Enter a number (0 - 3) in words:")

if num == "zero":
    print(0)
elif num == "one":
    print(1)
elif num == "two":
    print(2)
elif num == "three":
    print(3)
else:
    print("I don't know this number!")
```

### 10. 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!")
```

### 11. Solution

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

if wind < 0:
    print("Entered value is negative")
elif wind < 1:
    print("Beaufort: 0\n", "Calm", sep = "")
elif wind < 4:
    print("Beaufort: 1\n", "Light air", sep = "")
elif wind < 8:
    print("Beaufort: 2\n", "Light breeze", sep = "")
elif wind < 13:
    print("Beaufort: 3\n", "Gentle breeze", sep = "")
elif wind < 18:
    print("Beaufort: 4\n", "Moderate breeze", sep = "")
elif wind < 25:
    print("Beaufort: 5\n", "Fresh breeze", sep = "")
elif wind < 31:
    print("Beaufort: 6\n", "Strong breeze", sep = "")
elif wind < 39:
    print("Beaufort: 7\n", "Moderate gale", sep = "")
elif wind < 47:
    print("Beaufort: 8\n", "Gale", sep = "")
elif wind < 55:
    print("Beaufort: 9\n", "Strong gale", sep = "")
elif wind < 64:
    print("Beaufort: 10\n", "Storm", sep = "")
elif wind < 74:
    print("Beaufort: 11\n", "Violent storm", sep = "")
else:
    print("Beaufort: 12\n", "Hurricane force", sep = "")
```

### 12. 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: "))

if choice == 1:
    kelvin = float(input("Enter a temperature in degrees Kelvin: "))
    fahrenheit = 1.8 * kelvin - 459.67
    print(kelvin, "degrees Kelvin =", fahrenheit, "degrees Fahrenheit")
elif choice == 2:
    fahrenheit = float(input("Enter a temperature in degrees Fahrenheit: "))
    kelvin = (fahrenheit + 459.67) / 1.8
    print(fahrenheit, "degrees Fahrenheit =", kelvin, "degrees Kelvin")
elif choice == 3:
    fahrenheit = float(input("Enter a temperature in degrees Fahrenheit: "))
    celsius = 5 / 9 * (fahrenheit -32)
    print(fahrenheit, "degrees Fahrenheit =", celsius, "degrees Celsius")
elif choice == 4:
    celsius = float(input("Enter a temperature in degrees Celsius: "))
    fahrenheit = 9 / 5 * celsius + 32
    print(celsius, "degrees Celsius =", fahrenheit, "degrees Fahrenheit")
else:
    print("Invalid choice!")
```

# Chapter 16

## 16.2 Review Questions: True/False

1. True
2. True
3. False

## 16.3 Review Exercises

### 1. Solution

i. 25 6 ii. 10 9 iii. 50 2

### 2. 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)
```

### 3. 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")
```

# Chapter 17

## 17.3 Review Questions: True/False

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

# Chapter 18

## 18.2 Review Questions: True/False

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

## 18.3 Review Questions: Multiple Choice

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

## 18.4 Review Exercises

### 1. *Solution*

```
i = 30.0
while i > 5:
   print(i)
   i /= 2
print("The end")
```

### 2. *Solution*

Four

### 3. *Solution*

Zero

### 4. *Solution*

It displays

2 14 6

and performs three iterations

### 5. *Solution*

i. -1

ii. 9

iii. 0.5

iv. -7

v. A value between 17 and 32

vi. 1.4

### 6. Solution

```
total = 0

i = 1
while i <= 20:
    x = float(input("Enter a number: "))
    if x > 0:
        total += x
    i += 1
print(total)
```

### 7. Solution

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

p = 1
i = 1
while i <= n :
    x = float(input("Enter a number: "))
    if x > 0:
        p *= x
    i += 1
print(p)
```

### 8. Solution

```
total = 0
i = 1
while i <= 10 :
    x = int(input("Enter an integer: "))
    if x >= 100 and x <= 200:
        total += x
    i += 1
print(total)
```

### 9. Solution

```
total = 0
i = 1
while i <= 20 :
    x = int(input("Enter an integer between: "))
```

```
    if x >= 100 and x <= 999:
        total += x
    i += 1

print(total)
```

#### 10. Solution

```
p = 1
x = float(input("Enter a number: "))
while x != 0 :
    p *= x
    x = float(input("Enter a number: "))

print(p)
```

# Chapter 19

## 19.2 Review Questions: True/False

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

## 19.3 Review Questions: Multiple Choice

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

## 19.4 Review Exercises

#### 1. Solution

It displays

12 3

and performs five iterations

#### 2. Solution

It displays

10 4

19 20

28 32

#### 3. Solution

i. 9

ii. A value between 17 and 20

iii. -7 or -8

iv. -1

### 4. Solution

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

print(p, total / 20)
```

### 5. Solution

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

count = 0

for i in range(n):
    x = int(input("Enter an integer: "))
    if x > 0:
        count += 1

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

### 6. Solution

```
count_pos = 0
count_neg = 0

total_pos = 0
total_neg = 0

for i in range(50):
    x = int(input("Enter an integer: "))
    if x > 0:
        count_pos += 1
        total_pos += x
    elif x < 0:
        count_neg += 1
        total_neg += x
```

```
if count_pos > 0:
    print(total_pos / count_pos)

if count_neg > 0:
    print(total_neg / count_neg)
```

### 7. Solution

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

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

### 8. Solution

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

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

print(p)
```

### 9. Solution

First Approach

```
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")
```

Second Approach

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

count = 0
for character in msg:
    if character == " ":
```

```
        count += 1

words = count + 1

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

# Chapter 20

## 20.2 Review Questions: True/False

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

## 20.3 Review Questions: Multiple Choice

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

## 20.4 Review Exercises

### 1. Solution

i. 10

ii. A value between 46 and 50

iii. -7 or -8

iv. 137 or 138

### 2. Solution

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

### 3. Solution

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

### 4. Solution

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

### 5. Solution

```
for i in range(3):
    for j in range(8):
        print("* ", end = "")
    print()
```

### 6. Solution

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

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

### 7. Solution

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

### 8. Solution

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

for i in range(4, 0, -1):
    for j in range(i):
        print("* ", end = "")
    print()
```

# Chapter 21

## 21.7 Review Questions: True/False

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

## 21.8 Review Questions: Multiple Choice

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

## 21.9 Review Exercises

### 1. Solution

```
count_not_johns = 0
count_names = 0

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

print("Names other than John entered", count_not_johns, "times")
print(count_names, "names entered")
```

### 2. Solution

```
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")
```

### 3. Solution

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

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

if found == True:
    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

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

### 7. 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()
```

# Chapter 22

## 22.2 Review Exercises

### 1. Solution

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

print(total)
```

## 2. Solution

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

total = 0
for i in range(2, 2 * n + 1, 2):
    total += i

print(total)
```

## 3. Solution

```
n = int(input("Enter total number of students: "))

total = 0
count = 0
for i in range(n):
    grade = int(input("Enter grade: "))
    if grade >= 90 and grade <= 100:
        total += grade
        count += 1

if count > 0:
    print(total / count)
else:
    print("There are no students that got an A")
```

## 4. Solution

```
total = 0
count = 0
while total <= 3000:
    x = float(input("Enter a number: "))
    if x == 0:
        count += 1

    total += x

print(count)
```

## 5. Solution

```
answer = "YES"
while answer.upper() == "YES":
    r = float(input("Enter the length of a radius: "))

    area = 3.141 * r ** 2
    print("The area is:", area)
```

```
    answer = input("Would you like to repeat? ")
```

### 6. Solution

```
x = 1
while x <= 1073741824:
    print(x)
    x *= 2
```

### 7. Solution

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

### 8. 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"
```

### 9. Solution

```
t = float(input("Enter temperature for day 1: "))
maximum = t
total = t
for i in range(2, 32):
    t = float(input("Enter temperature for day " + str(i) + ": "))

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

print(total / 31, maximum)
```

### 10. Solution

```
level = float(input("Enter level: "))
hour = int(input("Enter hour: "))

maximum = minimum = level
```

```
max_hour = min_hour = hour

for i in range(23):
    level = float(input("Enter level: "))
    hour = int(input("Enter hour: "))

    if level > maximum:
        maximum = level
        max_hour = hour

    if level < minimum:
        minimum = level
        min_hour = hour


print(maximum, max_hour, minimum, min_hour)
```

### 11. Solution

```
import random

for i in range(2):
    secret_number = random.randrange(1, 101)

    attempts = 1

    guess = int(input("Enter a guess: "))
    while guess != secret_number:
        if guess > secret_number:
            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:
        attempts_1st_player = attempts
    else:
        attempts_2nd_player = attempts
```

```
if attempts_1st_player < attempts_2nd_player:
    print("First Player Wins")
elif attempts_2nd_player < attempts_1st_player:
    print("Second Player Wins")
else:
    print("It's a draw")
```

## 12. Solution

```
n = int(input("Enter total number of students: "))

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

for i in range(n):
    grade = int(input("Enter grade for student No" + str(i + 1) + ": "))

    gender = input("Enter gender for student No" + str(i + 1) + " (M/F): ")

    if 90 <= grade <= 100:
        total_a += grade
        count_a += 1
        if gender == "M":
            total_a_boys += grade
            count_a_boys += 1
    elif 80 <= grade <= 89:
        total_b += grade
        count_b += 1
    else:
        if gender == "F":
            count_cdef_girls += 1

    total += grade

if count_a > 0:
    print("Average value of those who got an 'A': ")
    print(total_a / count_a)
if count_b > 0:
    print("Average value of those who got a 'B': ")
```

```
    print(total_b / count_b)
if count_a_boys > 0:
    print("Average value of boys who got an 'A': ")
    print(total_a_boys / count_a_boys)

print("Total number of girls that got less than 'B':", count_cdef_girls)
print("Average grade of the whole class:", total / n)
```

### 13. Solution

```
answer = "yes"
while answer.upper() == "YES":
    amount = float(input("Enter amount: "))

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

    print("Discount:", discount, "%", sep = "")

    answer = input("Would you like to repeat? ")
```

# Chapter 23

## 23.12 Review Exercises

### 1. Solution

```
import turtle

wn = turtle.Screen()
george = turtle.Turtle()
george.shape("turtle")

george.forward(200)
george.left(130)
george.forward(50)

george.penup()
george.goto(200, 0)
```

```
george.pendown()

george.setheading(230)
george.forward(50)

wn.exitonclick()
```

### 2. Solution

```
import turtle

wn = turtle.Screen()
george = turtle.Turtle()
george.shape("turtle")

george.forward(200)
george.left(90 - 20)
george.forward(100)
george.left(90 + 20)
george.forward(200)
george.left(90 - 20)
george.forward(100)

wn.exitonclick()
```

### 3. Solution

```
import turtle

wn = turtle.Screen()
george = turtle.Turtle()
george.shape("turtle")

george.left(70)
george.forward(100)
george.left(40)
george.forward(100)
george.left(140)
george.forward(100)
george.left(40)
george.forward(100)

wn.exitonclick()
```

### 4. Solution

```
import turtle
```

```
wn = turtle.Screen()
george = turtle.Turtle()
george.shape("turtle")

george.penup()
george.goto(-200, 0)
george.pendown()

george.forward(300)
george.left(90 + 45)
george.forward(141)
george.left(45)
george.forward(100)
george.left(45)
george.forward(141)

wn.exitonclick()
```

### 5. Solution

```
import turtle

wn = turtle.Screen()
george = turtle.Turtle()
george.shape("turtle")

for k in range(2):
    for i in range(4):
        george.forward(100)
        george.left(90)

    george.penup()
    george.forward(100)
    george.pendown()

wn.exitonclick()
```

### 6. Solution

```
import turtle

wn = turtle.Screen()
george = turtle.Turtle()
george.shape("turtle")
```

```
for m in range(2):
    for n in range(2):
        for i in range(4):
            george.forward(100)
            george.left(90)

        george.penup()
        george.forward(200)
        george.pendown()

    george.penup()
    george.goto(0, 150)
    george.pendown()

wn.exitonclick()
```

## 7. Solution

```
import turtle

size = int(input("Enter pen size: "))
length = int(input("Enter length: "))
height = int(input("Enter height: "))

wn = turtle.Screen()
george = turtle.Turtle()
george.shape("turtle")

george.pensize(size)

george.forward(length)
george.left(90)
george.forward(height)
george.left(90)
george.forward(length)
george.left(90)
george.forward(height)

wn.exitonclick()
```

## 8. Solution

```
import turtle

length = int(input("Enter length of the size: "))
```

```
wn = turtle.Screen()
george = turtle.Turtle()
george.shape("turtle")


george.forward(length)
george.left(120)
george.forward(length)
george.left(120)
george.forward(length)
george.left(120)
george.forward(length)

wn.exitonclick()
```

### 9. Solution

```
import turtle

wn = turtle.Screen()
george = turtle.Turtle()
george.shape("turtle")

for i in range(12):
    george.forward(100)
    george.penup()
    george.backward(100)
    george.left(30)
    george.pendown()

wn.exitonclick()
```

### 10. Solution

```
import turtle

wn = turtle.Screen()
george = turtle.Turtle()
george.shape("turtle")

for k in range(0, 180, 60):
    for i in range(5):
        george.forward(150 + k)
        george.right(180 / 5 * 4)
```

```
    george.penup()
    george.backward(30)
    george.left(90)
    george.forward(10)
    george.right(90)
    george.pendown()

wn.exitonclick()
```

### 11. Solution

```
import turtle

wn = turtle.Screen()
george = turtle.Turtle()
george.shape("turtle")

for k in range(3):
    for i in range(4):
        george.forward(100)
        george.left(90)

    george.left(30)

wn.exitonclick()
```

### 12. Solution

```
import turtle

wn = turtle.Screen()
george = turtle.Turtle()
george.shape("turtle")

for k in range(12):
    for i in range(4):
        george.forward(100)
        george.left(90)

    george.left(30)

wn.exitonclick()
```

### 13. Solution

```
import turtle
```

```
wn = turtle.Screen()
george = turtle.Turtle()
george.shape("turtle")

for k in range(8):
    for i in range(4):
        george.forward(100)
        george.left(90)

    george.left(45)

wn.exitonclick()
```

## 14. Solution

```
import turtle

wn = turtle.Screen()
george = turtle.Turtle()
george.shape("turtle")
george.pensize(3)

#Poll position
george.penup()
george.goto(-300, 0)
george.pendown()

#Draw a blue rectangle
george.color("blue")
george.forward(200)
george.left(90)
george.forward(100)
george.left(90)
george.forward(200)
george.left(90)
george.forward(100)

#Move George to the top left corner of the rectangle
george.penup()
george.backward(100)
george.pendown()

#Draw the red roof
george.setheading(45)
george.color("red")
```

```
george.forward(141)
george.right(90)
george.forward(141)

#Draw the windows
george.color("brown")
george.penup()
george.setheading(0)
george.backward(180)
george.right(90)
george.forward(50)
george.left(90)
george.pendown()


for k in range(2):
    for i in range(4):
        george.forward(40)
        george.left(90)

    george.penup()
    george.forward(20)
    george.left(90)
    george.pendown()
    george.forward(40)

    george.penup()
    george.backward(20)
    george.right(90)
    george.backward(20)
    george.pendown()
    george.forward(40)

    george.penup()
    george.forward(80)
    george.right(90)
    george.forward(20)
    george.left(90)
    george.pendown()

#Draw the door
george.penup()
george.backward(180)
```

```
george.right(90)
george.forward(50)
george.left(90)
george.pendown()

george.forward(40)
george.left(90)
george.forward(70)
george.left(90)
george.forward(40)
george.left(90)
george.forward(70)

wn.exitonclick()
```

### 15. Solution

```
import turtle

wn = turtle.Screen()
george = turtle.Turtle()
george.shape("turtle")
george.pensize(3)

#Poll position
george.penup()
george.goto(-300, 0)
george.pendown()

for m in range(3):
    #Draw a blue rectangle
    george.color("blue")
    george.forward(200)
    george.left(90)
    george.forward(100)
    george.left(90)
    george.forward(200)
    george.left(90)
    george.forward(100)

    #Move George to the top left corner of the rectangle
    george.penup()
    george.backward(100)
    george.pendown()
```

```
#Draw the red roof
george.setheading(45)
george.color("red")
george.forward(141)
george.right(90)
george.forward(141)

#Draw the windows
george.color("brown")
george.penup()
george.setheading(0)
george.backward(180)
george.right(90)
george.forward(50)
george.left(90)
george.pendown()


for k in range(2):
    for i in range(4):
        george.forward(40)
        george.left(90)

    george.penup()
    george.forward(20)
    george.left(90)
    george.pendown()
    george.forward(40)

    george.penup()
    george.backward(20)
    george.right(90)
    george.backward(20)
    george.pendown()
    george.forward(40)

    george.penup()
    george.forward(80)
    george.right(90)
    george.forward(20)
    george.left(90)
    george.pendown()
```

```
    #Draw the door
    george.penup()
    george.backward(180)
    george.right(90)
    george.forward(50)
    george.left(90)
    george.pendown()

    george.forward(40)
    george.left(90)
    george.forward(70)
    george.left(90)
    george.forward(40)
    george.left(90)
    george.forward(70)

    george.penup()
    george.left(90)
    george.forward(210)
    george.pendown()

wn.exitonclick()
```

# Chapter 24

## 24.16 Review Questions: True/False

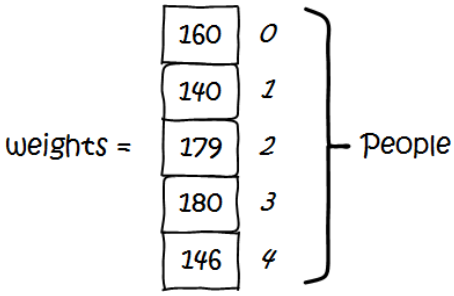
1. True
2. True
3. True
4. False
5. True
6. False
7. True
8. True
9. False
10. False
11. False
12. True
13. False
14. True
15. False
16. True
17. False
18. True
19. False
20. True
21. False
22. False
23. True
24. False
25. True
26. True
27. False
28. False
29. True
30. True
31. False
32. False
33. False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

## 24.17 Review Questions: Multiple Choice

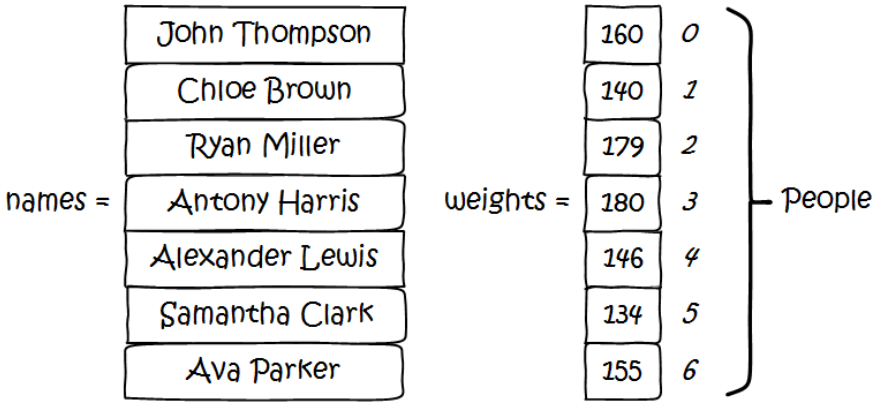
| | | | |
|---|---|---|---|
| 1. b | 6. d | 11. a | 16. a |
| 2. c | 7. c | 12. a | 17. a |
| 3. b | 8. b | 13. a | |
| 4. d | 9. b | 14. a | |
| 5. d | 10. b | 15. c | |

## 24.18 Review Exercises

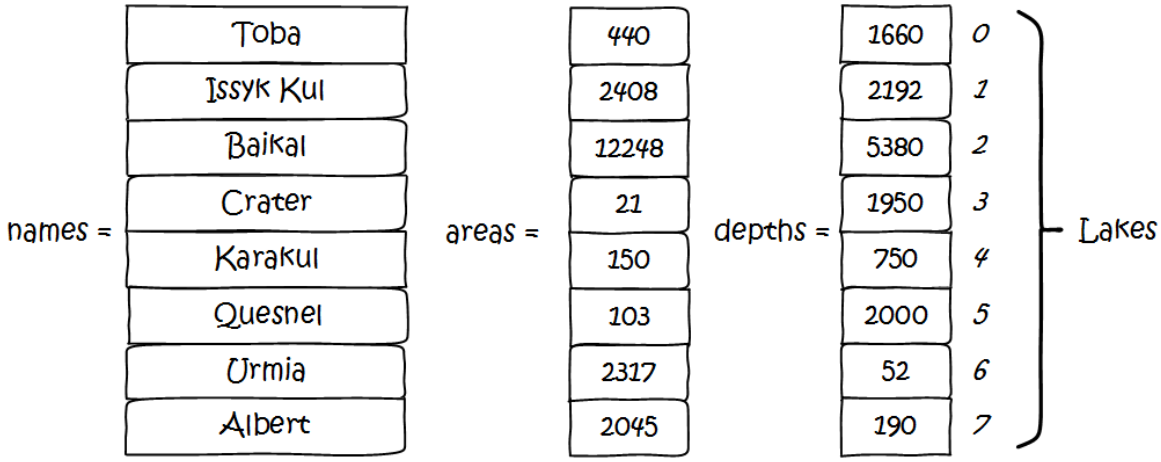
### 1. Solution



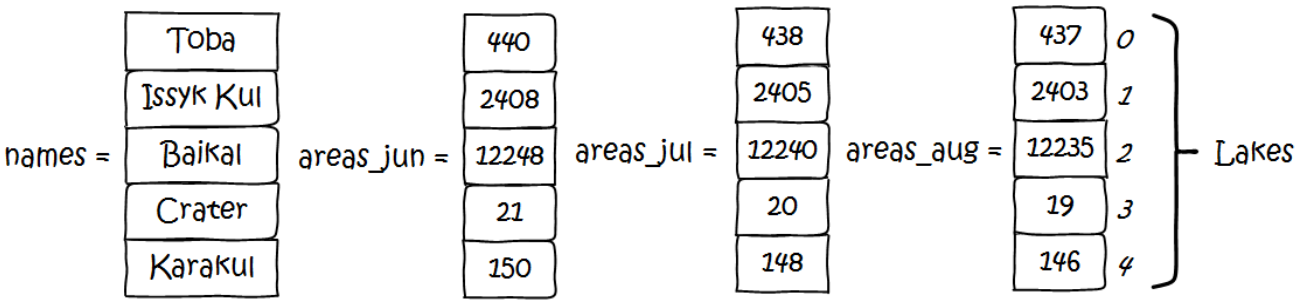

### 2. Solution



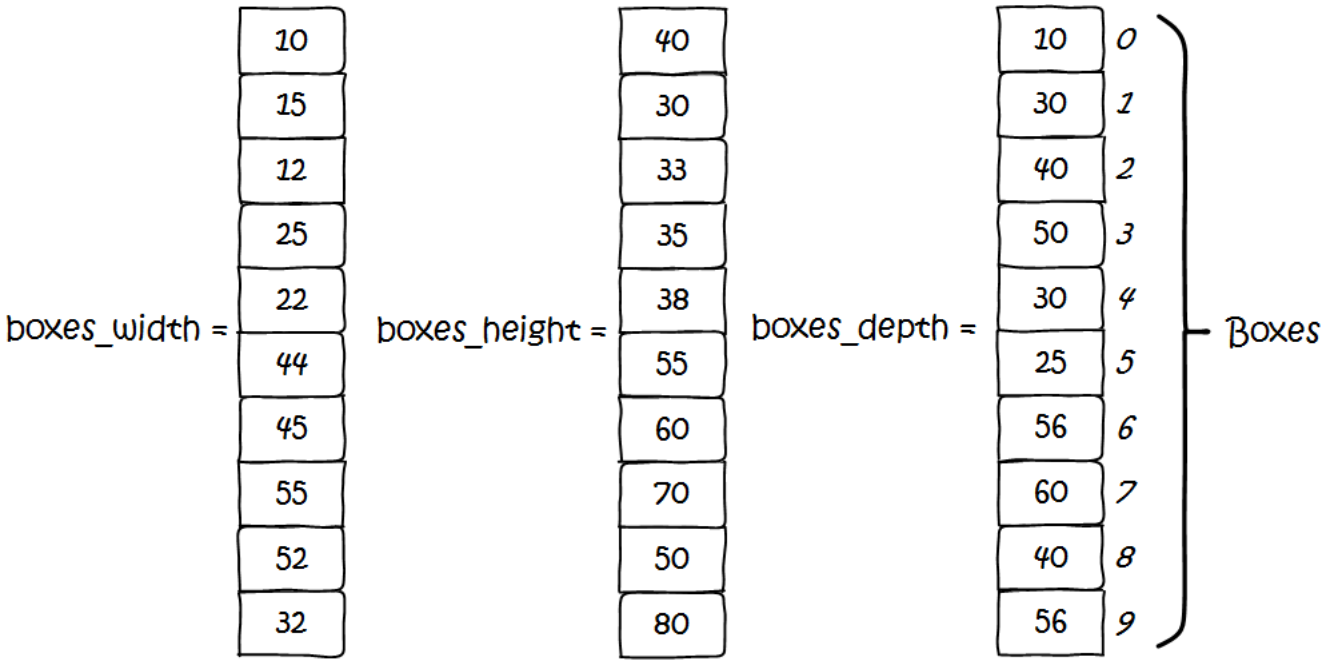

### 3. Solution



### 4. Solution




### 5. Solution




### 6. Solution

[16, 4, 1]

### 7. Solution

[4, 5, 10, 19, 10]

### 8. Solution

[18, 11, 46, 11, 11, 50]

### 9. Solution

[10, 22, 45, 67, 86, 19]

### 10. Solution

Navajo

Cherokee

Sioux

### 11. Solution

```
ELEMENTS = 100
```

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

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

### 12. Solution

```
ELEMENTS = 80

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

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

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

### 13. Solution

```
ELEMENTS = 50

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

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

### 14. Solution

```
ELEMENTS = 30

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

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

print(total)
```

### 15. Solution

```
ELEMENTS = 50

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

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

print(total)
```

### 16. Solution

```
ELEMENTS = 40

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

total_pos = 0
total_neg = 0
for element in a:
    if element > 0:
        total_pos += element
    elif element < 0:
        total_neg += element

print(total_pos, total_neg)
```

### 17. Solution

First approach

```
ELEMENTS = 20

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

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

```
print(total / ELEMENTS)
```

Second approach

```
import math
ELEMENTS = 20

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

print(math.fsum(a) / ELEMENTS)
```

### 18. Solution

```
ELEMENTS = 50

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

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

### 19. Solution

```
ELEMENTS = 60

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

for i in range(0, ELEMENTS, 2):
    print(a[i])
```

### 20. Solution

```
ELEMENTS = 20

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

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

print(total)
```

### 21. 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)
```

### 22. Solution

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

### 23. Solution

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

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

print(a)
```

### 24. 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)
```

### 25. 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)
```

### 26. Solution

```
ELEMENTS = 20

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

for word in words:
    if len(word) < 5:
        print(word)
```

### 27. Solution

```
ELEMENTS = 30

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

length_limits = (5, 10, 20)

for length_limit in length_limits:
    for word in words:
        if len(word) < length_limit:
            print(word)
```

### 28. Solution

```
ELEMENTS = 40

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

for word in words:
    count = 0
```

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

# Chapter 25

## 25.5 Review Questions: True/False

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

## 25.6 Review Exercises

### 1. Solution

First approach

```
ELEMENTS_OF_A = 50
ELEMENTS_OF_NEW = ELEMENTS_OF_A - 2

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

new_arr = [None] * ELEMENTS_OF_NEW
for i in range(ELEMENTS_OF_NEW):
    new_arr[i] = (a[i] + a[i + 1] + a[i + 2]) / 3

print(new_arr)
```

Second approach

```
import math
ELEMENTS_OF_A = 50

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

new_arr = []
for i in range(ELEMENTS_OF_A - 3):
    new_arr.append(math.fsum(a[i:i + 3]) / 3)

print(new_arr)
```

## 2. Solution

First approach

```
ELEMENTS = 15

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

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

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

new_arr = [None] * ELEMENTS
for i in range(ELEMENTS):
    minimum = a[i]
    if b[i] < minimum:
        minimum = b[i]
    if c[i] < minimum:
        minimum = c[i]
    new_arr[i] = minimum

print(new_arr)
```

Second approach

```
ELEMENTS = 15

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

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

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

new_arr = []
```

```
for i in range(ELEMENTS):
    new_arr.append(min(a[i], b[i], c[i]))

print(new_arr)
```

### 3. Solution

```
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]
index_of_max = 0
minimum = heights[0]
index_of_min = 0
for i in range(1, MOUNTAINS):
    if heights[i] > maximum:
        maximum = heights[i]
        index_of_max = i
    if heights[i] < minimum:
        minimum = heights[i]
        index_of_min = i

print(heights[index_of_max], names[index_of_max], countries[index_of_max])
print()
print(heights[index_of_min], names[index_of_min], countries[index_of_min])
```

### 4. Solution

```
CLASS1 = 20
CLASS2 = 25

print("Class A")
names1 = []
for i in range(CLASS1):
    names1.append(input("Enter name: "))

print("Class B")
names2 = []
for i in range(CLASS2):
```

```
    names2.append(input("Enter name: "))

needle = input("Enter a name to search: ")

found = False
for name in names1:
    if name == needle:
        found = True
        break

if found == True:
    print("Student found in class No 1")
else:
    found = False
    for name in names2:
        if name == needle:
            found = True
            break

    if found == True:
        print("Student found in class No 2")
    else:
        print("Student not found in either class")
```

### 5. Solution

```
usr = input("Enter username: ")
pwd = input("Enter password: ")

found = False
for i in range(100):
    if usernames[i] == usr:
        found = True
        break

if found == True:
    if usernames[i] == usr and passwords[i] == pwd:
        print("Login OK!")
    else:
        print("Login Failed!")
else:
    print("Login Failed!")
```

### 6. Solution

```
needle = input("Enter a value to search: ")
```

```
found = False
for i in range(1000):
    if SSNs[i] == needle:
        found = True
        print(SSNs[i], names[i])
        break

if found == False:
    for i in range(1000):
        if names[i] == needle:
            found = True
            print(SSNs[i], names[i])

if found == False:
    print("This value does not exist")
```

## 7. Solution

```
STUDENTS = 12

grades1 = []
grades2 = []
grades3 = []
for i in range(STUDENTS):
    grades1.append(int(input()))
    grades2.append(int(input()))
    grades3.append(int(input()))

found = False
for i in range(STUDENTS):
    if (grades1[i] + grades2[i] + grades3[i]) / 3 < 70:
        found = True
        break

if found == True:
    print("There is at least one student that has an average value below 70")
```

## 8. Solution

```
STUDENTS = 15

grades1 = []
grades2 = []
for i in range(STUDENTS):
    grades1.append(int(input()))
```

```
    grades2.append(int(input()))

for i in range(STUDENTS):
    print("Student No", (i + 1), ": ")

    average = (grades1[i] + grades2[i]) / 2

    if average < 60:
        print("E/F")
    elif average < 70:
        print("D")
    elif average < 80:
        print("C")
    elif average < 90:
        print("B")
    else:
        print("A")
```

## 9. Solution

```
PLAYERS = 15

points_match1 = []
points_match2 = []
points_match3 = []
points_match4 = []
for i in range(PLAYERS):
    points_match1.append(int(input()))
    points_match2.append(int(input()))
    points_match3.append(int(input()))
    points_match4.append(int(input()))

for i in range(PLAYERS):
    print("Player No", i + 1)
    print(points_match1[i] + points_match2[i] +    \
          points_match3[i] + points_match4[i])
```

## 10. Solution

```
HOURS = 24

t_city1 = []
t_city2 = []
t_city3 = []
for i in range(HOURS):
    t_city1.append(float(input()))
```

```
    t_city2.append(float(input()))
    t_city3.append(float(input()))

for i in range(HOURS):
    average = (t_city1[i] + t_city2[i] + t_city3[i]) / 3
    if average < 10:
        print("Hour:", (i + 1))
```

### 11. Solution

```
STUDENTS = 12

names = []
grd_lesson1 = []
grd_lesson2 = []
for i in range(STUDENTS):
    names.append(input())
    grd_lesson1.append(int(input()))
    grd_lesson2.append(int(input()))

#create a list average
average = []
for i in range(STUDENTS):
    average.append((grd_lesson1[i] + grd_lesson2[i]) / 2)

for i in range(STUDENTS):
    print(names[i], average[i])

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!")
```

### 12. Solution

```
ARTISTS = 15

artist_names = []
song_titles = []
scoreA = []
scoreB = []
scoreC = []
```

```
for i in range(ARTISTS):
    artist_names.append(input("Name for artist No." + str(i + 1) + ": "))
    song_titles.append( input("Song title for artist: " + artist_names[i]))
    print("Score for artist:", artist_names[i])
    scoreA.append(int(input(" gotten from judge A: ")))
    scoreB.append(int(input(" gotten from judge B: ")))
    scoreC.append(int(input(" gotten from judge C: ")))

total = []
for i in range(ARTISTS):
    minimum = min(scoreA[i], scoreB[i], scoreC[i])
    total.append(scoreA[i] + scoreB[i] + scoreC[i] - minimum)

for i in range(ARTISTS):
    print(artist_names[i], song_titles[i], total[i])
```

### 13. Solution

```
CITIZENS = 20

answers1 = []
answers2 = []

prod_name1 = input("Enter Product Name 1: ")
for i in range(CITIZENS):
    answers1.append(input("Enter score for product " + prod_name1 + ": "))

prod_name2 = input("Enter Product Name 2:")
for i in range(CITIZENS):
    answers2.append(input("Enter score for product " + prod_name2 + ": "))

count_A = 0
for i in range(CITIZENS):
    if answers1[i] == "A":
        count_A += 1
print(prod_name1, count_A)

count_A = 0
for i in range(CITIZENS):
    if answers2[i] == "A":
        count_A += 1
print(prod_name2, count_A)
```

### 14. 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 a word: ")

for letter in word:
    print(morseAlphabet[letter.upper()], end = " ")
```

# Chapter 26

## 26.4 Review Questions: True/False

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

# Chapter 27

## 27.11 Review Questions: True/False

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

## 27.12 Review Exercises

### 1. Solution

```
def find_max(a, b):
    if a > b:
        maximum = a
    else:
        maximum = b
    return maximum
```

### 2. Solution

It displays:

3 is positive

-7 is negative or zero

-9 is negative or zero

0 is negative or zero

4 is positive

### 3. Solution

```
def find_sum(a, b, c):
    return a + b + c
```

### 4. Solution

```
def find_avg(a, b, c, d):
    return (a + b + c + d) / 4
```

### 5. Solution

```
def display_max(a, b, c):
    print(max(a, b, c))
```

## 6. Solution

```
def find_min(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())

#Display lowest value as follows (1st approach)
temp1 = find_min(x1, x2)
temp2 = find_min(x3, x4)
print(find_min(temp1, temp2))

#Or as follows (2nd approach)
print(find_min(find_min(x1, x2), find_min(x3, x4)))
```

## 7. Solution

```
def get_input():
    answer = input("Enter Yes or No: ")
    if answer.upper() == "YES":
        return True
    else:
        return False

def find_area(b, h):
    return b * h

#Main code starts here
answer = True
while answer == True:
    bas = float(input("Enter the base of the parallelogram: "))
    height = float(input("Enter the height of the parallelogram: "))

    print("Area =", find_area(bas, height))

    print("Would you like to repeat? ")
    answer = get_input()
```

# Chapter 28

## 28.2 Review Exercises

### 1. Solution

```
def Kelvin_to_Fahrenheit(kelvin):
    return 1.8 * kelvin - 459.67

def Kelvin_to_Celsius(kelvin):
    return kelvin - 273.15

#Main code starts here
k = float(input("Enter a temperature in degrees Kelvin: "))
print("Fahrenheit:", Kelvin_to_Fahrenheit(k))
print("Celsius:", Kelvin_to_Celsius(k))
```

### 2. Solution

```
def num_of_days(month):
    if month in [4, 6, 9, 11]:
        days = 30
    elif month == 2:
        days = 28
    else:
        days = 31

    return days

#Main code starts here
x = int(input("Enter a month: "))
y = int(input("Enter a second month: "))

total = 0
for i in range(x, y + 1):
    total += num_of_days(i)

print(total)
```

### 3. Solution

```
import random

def dice():
    return random.randrange(1, 7)
```

```
#Main code starts here
names = []
names.append(input("Player 1 enter name: "))
names.append(input("Player 2 enter name: "))

total = [0, 0]

for player in range(2):
    for i in range(10):
        print(names[player], ", hit enter to roll the dice!")
        key = input()

        dice1 = dice()
        dice2 = dice()
        print(dice1, dice2)
        total[player] += dice1 + dice2

if total[0] == total[1]:
    print("Tie!")
elif total[0] > total[1]:
    print(names[0], "wins!")
else:
    print(names[1], "wins!")
```

### 4. Solution

```
def bmi(w, h):
    b = w * 703 / h ** 2
    if b < 16:
        print("You must add weight.")
    elif b < 18.5:
        print("You should add some weight.")
    elif b < 25:
        print("Maintain your weight.")
    elif b < 30:
        print("You should lose some weight.")
    else:
        print("You must lose weight.")

#Main code starts here
weight = float(input("Enter your weight (in pounds): "))
age = int(input("Enter your age: "))
height = float(input("Enter your height (in inches): "))

if age < 18:
```

```
    print("I can't calculate your BMI. You must be adult!")
else:
    bmi(weight, height)
```

## 5. Solution

```
CARS = 40
GAS = 1
DIESEL = 2
HYBRID = 3

def get_choice():
    print("1. Gas")
    print("2. Diesel")
    print("3. Hybrid")
    choice = int(input("Enter type of the car: "))
    return choice

def get_days():
    days = int(input("Enter total number of rental days: "))
    return days

def get_charge(car_type, rental_days):
    if car_type == GAS:
        if rental_days <= 5:
            charge = rental_days * 24
        else:
            charge = rental_days * 22
    elif car_type == DIESEL:
        if rental_days <= 5:
            charge = rental_days * 28
        else:
            charge = rental_days * 25
    else:
        if rental_days <= 5:
            charge = rental_days * 30
        else:
            charge = rental_days * 28

    return charge

#Main code starts here
rented_car_types = [None] * CARS
rented_days = [None] * CARS
```

```
for i in range(CARS):
    rented_car_types[i] = get_choice()
    rented_days[i] = get_days()

total = 0
for i in range(CARS):
    charge = get_charge(rented_car_types[i], rented_days[i])
    print("Amount to pay, car No", (i + 1), ":", charge)
    total += charge

count = 0
for i in range(CARS):
    if rented_car_types[i] == HYBRID:
        count += 1

print("Hybrids rented:", count)

print("Total profit:", total)
```

# Chapter 29

## 29.9 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

## 29.10 Review Exercises

### *1. Solution*

```
class Trigonometry:
    def square_area(self, side):
        return side * side

    def rectangle_area(self, base, height):
        return base * height

    def triangle_area(self, base, height):
        return base * height / 2

#Main code starts here
```

```
tr = Trigonometry()

sqr_side = float(input("Enter square side: "))

rctngl_base = float(input("Enter rectangle base: "))
rctngl_height = float(input("Enter rectangle height: "))

trngl_base = float(input("Enter triangle base: "))
trngl_height = float(input("Enter triangle height: "))

print(tr.square_area(sqr_side))
print(tr.rectangle_area(rctngl_base, rctngl_height))
print(tr.triangle_area(trngl_base, trngl_height))
```

## 2. Solution

```
class Pet:
    def __init__(self):
        self.kind = None
        self.legs_number = None

    def start_running(self):
        print("Pet is running")

    def stop_running(self):
        print("Pet stopped")

#Main code starts here
pet1 = Pet()
pet1.kind = "dog"
pet1.legs_number = 4

pet2 = Pet()
pet2.kind = "monkey"
pet2.legs_number = 2

pet1.start_running()
pet2.start_running()
pet1.stop_running()
```

## 3. Solution

```
class Pet:
    def __init__(self, kind, legs_number):
        self.kind = kind
        self.legs_number = legs_number
```

```
    #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 legs_number(self):
        return self._legs_number

    #Define the setter
    @legs_number.setter
    def legs_number(self, value):
        if value >= 0:
            self._legs_number = value
        else:
            raise ValueError("Cannot be negative")

    def start_running(self):
        print("Pet is running")

    def stop_running(self):
        print("Pet stopped")

#Main code starts here
pet1 = Pet("dog", 4)

pet1.start_running()
pet1.stop_running()

pet1.kind = ""          #this will throw an error
pet1.legs_number = -3   #this will throw an error
```

## 4. Solution

```
BOXES = 3
```

```
class Box:
    def __init__(self, width, length, height):
        self.width = width
        self.length = length
        self.height = height

    def display_volume(self):
        print("Volume", self.width * self.length * self.height)

    def display_dimensions(self):
        print(self.width, "x", self.length, "x", self.height)

#Main code starts here
list_of_obj = [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
    list_of_obj[i] = Box(w, l, h)

for i in range(BOXES):
    list_of_obj[i].display_dimensions()
    list_of_obj[i].display_volume()
```

## 5. Solution

```
BOXES = 3

class Box:
    def __init__(self, width, length, height):
        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 display_volume(self):
        print("Volume", self.width * self.length * self.height)

    def display_dimensions(self):
        print(self.width, "x", self.length, "x", self.height)

#Main code starts here
list_of_obj = [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
    list_of_obj[i] = Box(w, l, h)

for i in range(BOXES):
    list_of_obj[i].display_dimensions()
    list_of_obj[i].display_volume()
```

## 6. Solution

```
class Cube:
    def __init__(self, edge):
        self.edge = edge

    def display_volume(self):
        print("Volume:", self.edge ** 3)

    def display_one_surface(self):
        print("One surface:", self.edge ** 2)

    def display_total_surface(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.display_volume()
cube1.display_one_surface()
cube1.display_total_surface()
```

## 7. Solution

```
class Cube:
    def __init__(self, edge):
        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 display_volume(self):
        print("Volume:", self.edge ** 3)

    def display_one_surface(self):
        print("One surface:", self.edge ** 2)

    def display_total_surface(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.display_volume()
cube1.display_one_surface()
cube1.display_total_surface()
```

## 8. Solution

```
def display_menu():
    print("1. Enter radius")
    print("2. Display radius")
    print("3. Display diameter")
    print("4. Display area")
    print("5. Display perimeter")
    print("6. Exit")

class Circle:
    def __init__(self):
        self._radius = None   #private field

    #Define the getter
    @property
    def radius(self):
        if self._radius != None:
            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 get_diameter(self):
        return 2 * self.radius

    def get_area(self):
        return 3.14 * self.radius ** 2

    def get_perimeter(self):
        return 2 * 3.14 * self.radius

#Main code starts here
circle1 = Circle()

while True:
    display_menu()

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

    if choice == 6:
        print("Bye")
        break
    elif choice == 1:
        radius = float(input("Enter radius: "))
        circle1.radius = radius
    elif choice == 2:
        print("Radius:", circle1.radius)
    elif choice == 3:
        print("Diameter:", circle1.get_diameter())
    elif choice == 4:
        print("Area:", circle1.get_area())
    else:
        print("Perimeter:", circle1.get_perimeter())
```

## 9. Solution

```
class Info:

    def __init__(self):
        self._user_text = None   #Private field. It does not call the setter!

    #Define the getter
    @property
    def user_text(self):
        return self._user_text

    #Define the setter
    @user_text.setter
    def user_text(self, value):
        if value != "":
            self._user_text = value
        else:
            raise ValueError("Cannot be set to empty")

    def get_spaces_count(self):
        count = 0
        for char in self.user_text:
            if char == " ":
                count += 1
        return count

    def get_words_count(self):
        return self.get_spaces_count() + 1

    def get_vowels_count(self):
        count = 0
        for char in self.user_text:
            if char in "aeiou":
                count += 1
        return count

    def get_letters_count(self):
        return len(self.user_text) - self.get_spaces_count()

#Main code starts here
inf = Info()

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

```
inf.user_text = text

print("Text:", inf.user_text)
print("Spaces:", inf.get_spaces_count())
print("Words:", inf.get_words_count())
print("Vowels:", inf.get_vowels_count())
print("Total number of letters:", inf.get_letters_count())
```

### 10. Solution

```
def display_menu():
    print("1. Encryption/Decryption key")
    print("2. Encrypt a message")
    print("3. Decrypt a message")
    print("4. Exit")

class EncryptDecrypt:

    def __init__(self):
        self._encr_decr_key = None  #Private field.
                                    #It does not call the setter!

        self._alphabet = " abcdefghijklmnopqrstuvwxyz"  # space is
                                                        # a valid character!
    #Define the getter
    @property
    def encr_decr_key(self):
        if self._encr_decr_key != None:
            return self._encr_decr_key
        else:
            raise ValueError("Key is not set")

    #Define the setter
    @encr_decr_key.setter
    def encr_decr_key(self, value):
        if value in range(1,27):
            self._encr_decr_key = value
        else:
            raise ValueError("Key must be between 1 and 26")

    def encrypt(self, message):
        return_value = ""
        for char in message:
            index = self._alphabet.find(char)
```

```
            new_index = index + self.encr_decr_key
            if new_index >= 27:
                new_index -= 27
            new_letter = self._alphabet[new_index]
            return_value += new_letter
        return return_value

    def decrypt(self, enc_message):
        return_value = ""
        for char in enc_message:
            index = self._alphabet.find(char)
            new_index = index - self.encr_decr_key
            if new_index < 0:
                new_index += 27
            new_letter = self._alphabet[new_index]
            return_value += new_letter
        return return_value

#Main code starts here
ed = EncryptDecrypt()

while True:
    display_menu()

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

    if choice == 4:
        print("Bye")
        break
    elif choice == 1:
        encr_decr_key = int(input("Enter encryption/decryption key: "))
        ed.encr_decr_key = encr_decr_key
    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))
```