German University in Cairo
Faculty of Media Engineering and Technology
Prof. Dr. Slim Abdennadher
Dr. Aysha ElSafty

Introduction to Computer Science, Winter Semester 2016
Practice Assignment 3
Discussion: 22.10.2016 - 27.10.2016

Exercise 3-1

Discount

A discount is made on a purchase as follows:

- if purchase $\leq$ 1000 L.E., there is no discount
- if purchase $>$ 1000 L.E., the discount is 5%

Given the cost of the purchase, write an algorithm to calculate and print the money paid taking into consideration the 10% sales taxes. The taxes are calculated on the amount after the discount.

Solution:

```
purchase = eval(input())
if(purchase <= 1000):
␣ print(" There is no discount! ")
else:
␣ discount = purchase * 0.05
␣ purchase = purchase - discount

amount = purchase + purchase * 0.1
print(" The amount due is: ")
print(amount)
```

Exercise 3-2

Gas Station

National Last Chance gas station sits on Cairo Alexandria route. There is no other gas station for 200 Miles. Write an algorithm to help drivers decide if they need gas or not. The user will enter:

- The capacity of the gas tank, in Gallons.
- The indication of the gas gauge in percent (full=100, three quarters=75 and so on).
- The miles per gallon of the car.

The following is a sample run of the algorithm:

```
Tank Capacity:                12
Gas Gauge Reading in percent: 50
Miles per Gallon:             30

Get Gas!
```

The algorithm should print out `Get gas` or `Safe to proceed` depending on if the car can cross the 200 miles with the gas remaining in the tank.

Solution:

```
t = eval(input())      # Tank Capacity
g = eval(input())     # Gage Reading Percent
m = eval(input())     # Miles per gallon
ng = (t * g) / 100  # Number of gallons in tank
nm = ng * m     # Number of miles that can be made
if(nm >= 200):
␣ print("Safe To Proceed")
else:
␣ print("Get Gas !")
```

Exercise 3-3

BMI

Write an algorithm that determines whether you are underweight, fit or overweight given your **weight** and **height** based on your BMI calculation.

The BMI is calculated using the weight divided by height squared, where weight is in kg and height is in meters.

- If BMI ≤ 18.5, you are underweight
- If BMI > 18.5 and BMI ≤ 25, you are fit
- If BMI > 25, you are overweight

Solution:

-
  ```
  weight, height = eval(input("Please enter weight")),eval(input("Please enter height"))

  BMI = weight/(height**2)

  if(BMI <= 18.5):
  ␣ print("You are underweight")
  elif(BMI <= 25):
  ␣ print("You are fit")
  else:
  ␣ print("You are overweight")
  ```

-
  ```
  weight, height = eval(input("Please enter weight")),eval(input("Please enter height"))

  BMI = weight/(height**2)

  if(BMI <= 18.5):
  ␣ print("You are underweight")
  else:
  ␣ if(BMI <= 25):
  ␣␣ print("You are fit")
  ␣ else:
  ␣␣ print("You are overweight")
  ```

Exercise 3-4

Triangle

Write an algorithm that determines whether A, B, and C can form the sides of a triangle. Note A, B and C can form the sides of a triangle if each side is less than the sum of the two other sides, i.e.: $A < B + C, B < A + C$ and $C < A + B$.

If A, B, and C forms a triangle, calculate its area using the formula:

$$Area = \sqrt{S(S-A)(S-B)(S-C)}\ , \quad \text{where } S = (A+B+C)/2$$

Solution:

-
  ```
  import math
  A,B,C = eval(input()), eval(input()), eval(input())

  if(A < B + C):       # first if
  ␣ if(B < C+ A):     #second if
  ␣␣ if(C < A + B):  #third if
  ␣␣␣ S = ((A + B+ C) / 2)
  ␣␣␣ Area = (math.sqrt(S * (S - A) * ( S - B) * (S - C)))
  ␣␣␣ print(" This is a triangle with the Area: ")
  ␣␣␣ print(Area)
  ␣␣ else:                  #else of 3rd if
  ␣␣␣ print("This cannot be a triangle!")

  ␣ else:                     #else of 2nd if
  ␣␣ print("This cannot be a triangle!")

  else:                       #else of 1st if
  ␣ print("This cannot be a triangle!")
  ```

-
  ```
  import math
  A, B, C = eval(input()), eval(input()), eval(input())
  triangle = 0
  if(A < B + C):
  ␣ if(B < C+ A):
  ␣␣ if(C < A + B):
  ␣␣␣ triangle = 1

  if(triangle == 0):
  ␣ print("This cannot be a triangle!")
  else:
  ␣ S = ((A + B+ C) / 2)
  ␣ Area = (math.sqrt(S * (S - A) * ( S - B) * (S - C)))
  ␣ print(" This is a triangle with the Area: ")
  ␣ print(Area)
  ```

-
  ```
  import math
  A, B, C = eval(input()), eval(input()), eval(input())

  if(A < B + C and B < C+ A and C < A + B):
  ␣ S = ((A + B+ C) / 2)
  ␣ Area = (math.sqrt(S * (S - A) * ( S - B) * (S - C)))
  ␣ print(" This is a triangle with the Area: ")
  ```

```
␣ print(Area)
else:
␣ print("This cannot be a triangle!")
```

Exercise 3-5

Letter Grades

Students marks in a class are graded on the following policy:

- A: 85-100
- B: 74-85
- C: 60-74
- D: 50-60
- F: <50

Write an algorithm that reads each student's marks, print either a grade or an error message.

Solution:

-
  ```
  Mark = eval(input())
  if(Mark < 0):
  ␣ print("invalid mark, less than 0")
  else:
  ␣ if(Mark >105):
  ␣␣ print("invalid mark, greater than 105")
  ␣ else:
  ␣␣ if(Mark <50):
  ␣␣␣ print("grade is F")
  ␣␣ else:
  ␣␣␣ if(Mark <60):
  ␣␣␣␣ print("grade is D")
  ␣␣␣ else:
  ␣␣␣␣ if(Mark <74):
  ␣␣␣␣␣ print("grade is C")
  ␣␣␣␣ else:
  ␣␣␣␣␣ if(Mark <85):
  ␣␣␣␣␣␣ print("grade is B")
  ␣␣␣␣␣ else:
  ␣␣␣␣␣␣ print("grade is A")
  ```

-
  ```
  Mark = eval(input())
  if(Mark < 0):
  ␣ print("invalid mark, less than 0")
  elif(Mark >105):
  ␣ print("invalid mark, greater than 105")
  elif(Mark <50):
  ␣ print("grade is F")
  elif(Mark <60):
  ␣ print("grade is D")
  elif(Mark <74):
  ␣ print("grade is C")
  elif(Mark <85):
  ␣ print("grade is B")
  else:
  ```

```
    ␣ print("grade is A")
```

Exercise 3-6

Student School

The following algorithm prints out whether a current student is in elementary (1st - 5th), middle (6th - 8th), or high school (9th - 12th).

```
grade = eval(input())
if(grade <= 5):
␣ print("this student is in elementary school")
elif(grade <= 8):
␣ print("this student is in middle school")
elif(grade <= 12):
␣ print("this student is in high school")
```

The algorithm above uses nested if-statements.

a) Write an equivalent algorithm that will print the same messages as the algorithm above **without using any nested if-statements**.

   Solution:

```
grade = eval(input())
if(grade <= 5):
␣ print("this student is in elementary school")

if(grade <= 8 and grade > 5):
␣ print("this student is in middle school")

if(grade <= 12 and grade > 8):
␣ print("this student is in high school")
```

b) Discuss the drawback of your algorithm? **Hint:** Compare the efficiency of both algorithms.

   Solution:

   The first algorithm is more efficient than the second one since in the second algorithms all conditions should be checked. In the first algorithm the number of conditions that should be checked depend on the input.

Exercise 3-7

Balance

Consider the following algorithm:

```
balance = eval(input())
if((balance >= 2000) or (balance <= 3000)):
␣ print("Your balance is between 2000$ and 3000$")
␣ print("Your interest rate will be 3.5%")
else:
␣ print("Your balance is larger than 3000$")
␣ print("Your interest rate will be 4.5%")
```

a) Do you think this algorithm does what the programmer intended? Justify your answer.

b) If the algorithm does not do what the programmer intends, improve this algorithm.

Solution:

The code above does not do what the programmer intended. We have two problems:

- The condition of the if statement should be an `and` instead of `or`.
- In the else part an if statement is missing that checks that the balance is larger than `3000`.

```
balance = eval(input())
if((balance >= 2000) and (balance <= 3000)):
␣ print("Your balance is between $2000 and $3000")
␣ print("Your interest rate will be 3.5%")
else:
␣ if(balance > 3000):
␣␣ print("Your balance is larger than $3000.")
␣␣ print("Your interest rate will be 4.5%")
```

Exercise 3-8

Mysterious Task

- What does the following program display for any boolean values x, y and z? Choose one answer from the below choices and Justify.

  ```
  x = eval(input())
  y = eval(input())
  z = eval(input())
  if (x and y):      #if (x == True and y == True)
  ␣ print((not x) or z)
  else:
  ␣ print((x and y) or z)
  ```

  a) The value of x
  b) The value of y
  c) The value of z
  d) The value of x and y
  e) Always true
  f) True if either x and y are both True or z is True, and False otherwise.

  Solution:

  The solution is the value of z.

  If x and y are true, then not x is false and the then part will display then the value of z. Otherwise, either x or y is false, then x and y will be false too, thus the else part will display the value of z too.

- Consider a program which changes the value of a boolean variable that tracks whether a light is on or off. Three people, creatively named Sarah, Ali, and Mina, who are claiming to be expert light switchers have written different implementations of this function.

  Sarah's Version:

  ```
  lightOn = eval(input())
  if (lightOn):
  ␣ lightOn = False
  else:
  ␣ lightOn = True
  ```

Ali's Version:

```
lightOn = eval(input())
if (lightOn):
␣ lightOn = False
if (not lightOn):
␣ lightOn = True
```

Minas's Version:

```
lightOn = eval(input())
lightOn = not lightOn
```

Are these implementations equivalent, i.e. performing the same task? Are all three people expert light switchers? Why? Justify your answer.

Solution:

No, these implementations are not equivalent. Sarah and Mina are experts, but Ali is a fraud. If the light is on, his code turns it off then the second condition will be true and will go to the then part, so it will always remain on.