

ITA6017 - PYTHON PROGRAMMING
ASSIGNMENT 1

REG NO: 22MCA0223

NAME: KAMRAN ANSARI

Question 1

Percentage of non-defective items

A bag contains a total of 'T' nuts and bolts. Out of which there are 'N' nuts. In a quality check, x% of the nuts and y% of bolts in the bag were found to be defective. Design an algorithm and write a Python code to determine the percentage of non-defective items in the bag. For example if T is 200, N is 150, x is 50 and y is 20 then the percentage of non-defective items in bag is 57.5. Round the answer to two decimal places using format function.

Input for the Problem

'T' – total number of nuts and bolts in the bag

'N' – Number of nuts

'x' – Percentage of defective nuts in the bag

'y' – Percentage of defective bolts in the bag

Output of the Problem

Percentage of non-defective items in bag

Boundary Conditions

All inputs > 0

Test Data

Input

200

150

50

20

\$END

300

100

40

15

Output

57.50

\$END

76.67

Code

```
print("Enter total number of items in the bag: ")
totalItems = int(input())

print("Enter number of nuts in the bag: ")
totalNuts = int(input())

bolts = totalItems - totalNuts

print("Enter percentage of defective nuts in the bag: ")
defNutsPercent = float(input()) / 100

print("Enter percentage of defective bolts in the bag: ")
defBoltsPercent = float(input()) / 100

defNuts = int(totalNuts * defNutsPercent)
defBolts = int(bolts * defBoltsPercent)

totalDef = defNuts + defBolts
totalNonDef = totalItems - totalDef

nonDefPercent = (totalNonDef / totalItems) * 100

print("Percentage of non-defective items in bag: {:.2f}".format(nonDefPercent))
```

Output

```
deadmercury@DESKTOP-QUKDS0U MINGW64 /e/LEARNING/vit_mca/ITA6017_PYTHON (main)
$ C:/Python311/python.exe e:/LEARNING/vit_mca/ITA6017_PYTHON/DA/DA_1/Q1.py
Enter total number of items in the bag:
200
Enter number of nuts in the bag:
150
Enter percentage of defective nuts in the bag:
50
Enter percentage of defective nuts in the bag:
20
Percentage of non-defective items in bag: 57.50

deadmercury@DESKTOP-QUKDS0U MINGW64 /e/LEARNING/vit_mca/ITA6017_PYTHON (main)
$ C:/Python311/python.exe e:/LEARNING/vit_mca/ITA6017_PYTHON/DA/DA_1/Q1.py
Enter total number of items in the bag:
300
Enter number of nuts in the bag:
100
Enter percentage of defective nuts in the bag:
40
Enter percentage of defective nuts in the bag:
15
Percentage of non-defective items in bag: 76.67
```

Question 2

Calculate the number of birthdays celebrated by Mr.X. If Mr.X was born on 29th February of a leap year then he celebrates birthday only in leap years. Given the date of birth of Mr.X and the current year, design an algorithm and write the Python code to determine the number of birthdays celebrated by Mr.X. A year is a leap year if it is divisible by 4 and not divisible by 100 or when the year is divisible by 400. For example, year 1996 and 2000 are leap years whereas 1900 is not a leap year. Assume that the current day and month is greater than day and month of birthday.

Input Format:

day of birth

month of birth

year of birth

Current year

Output Format:

Number of birthdays celebrated by Mr. 'X'

Boundary Condition:

Current Year > Year of birth

Input:

29

2

1960

2010

\$END

31

1

1975

2100

\$END

3

2

1960

1950

Output:

12

\$END

125

\$END

Invalid input

Code

```
def isLeapYear(year):  
    if (year % 4 == 0 and year % 100 != 0) or (year % 400 == 0):  
        return True  
    return False  
  
print("Enter date of birth: ")  
birthDate = int(input())  
  
print("Enter month of birth: ")  
birthMonth = int(input())  
  
print("Enter year of birth: ")  
birthYear = int(input())  
  
print("Enter current year: ")  
currentYear = int(input())  
  
if (currentYear < birthYear):  
    print("Invalid Input")  
    exit()  
  
if (birthDate == 29 and birthMonth == 2):  
    noOfBirthdays = 0  
  
    for i in range(birthYear + 1, currentYear):  
        noOfBirthdays += 1 if isLeapYear(i) else 0
```

```
        print("Number of birthdays celebrated: " + str(noOfBirthdays))
else:
    print("Number of birthdays celebrated: " + str(currentYear -
birthYear))
```

Output

```
deadmercury@DESKTOP-QUKDS0U MINGW64 /e/LEARNING/vit_mca/ITA6017_PYTHON (main)
$ C:/Python311/python.exe e:/LEARNING/vit_mca/ITA6017_PYTHON/DA/DA_1/Q2.py
Enter date of birth:
29
Enter month of birth:
2
Enter year of birth:
1960
Enter current year:
2010
Number of birthdays celebrated: 12
```

```
deadmercury@DESKTOP-QUKDS0U MINGW64 /e/LEARNING/vit_mca/ITA6017_PYTHON (main)
$ C:/Python311/python.exe e:/LEARNING/vit_mca/ITA6017_PYTHON/DA/DA_1/Q2.py
Enter date of birth:
31
Enter month of birth:
1
Enter year of birth:
1975
Enter current year:
2100
Number of birthdays celebrated: 125
```

```
deadmercury@DESKTOP-QUKDS0U MINGW64 /e/LEARNING/vit_mca/ITA6017_PYTHON (main)
$ C:/Python311/python.exe e:/LEARNING/vit_mca/ITA6017_PYTHON/DA/DA_1/Q2.py
Enter date of birth:
3
Enter month of birth:
2
Enter year of birth:
1960
Enter current year:
1950
Invalid Input
```


Question 3

A person 'X' has a farm which is rectangular in shape. 'X' wants to plant coconut trees in the farm and he has heard that every tree has to be separated by 2 feet. He needs to calculate the number of plants to be purchased. Write a program to calculate the number of plants when provided with the length and breadth of the farm (in feet). The program should display the number of rows and columns along with the total number of plants required for the farm. For example, if the length and breadth of the farm is 9 feet * 4 feet then the farmer can plant trees in positions 0, 2, 4, 6, 8 along the length and 0,2,4 along the breadth therefore number of trees to be purchased is $5 \times 3 = 15$.

Input Format:

The first input value read is the length of the farm (in feet).

The second input value read is the breadth of the farm (in feet).

Output Format:

Print number of rows

Print number of columns

Print number of plants required

Code

```
print("Enter length of farm: ")
farmLength = int(input())

print("Enter breadth of farm: ")
farmBreadth = int(input())

noOfPlantsAlongLength = 0
```

```
for i in range(0, farmLength + 1, 2):
    noOfPlantsAlongLength += 1

noOfPlantsAlongBreadth = 0

for i in range(0, farmBreadth + 1, 2):
    noOfPlantsAlongBreadth += 1

totalNoOfPlants = noOfPlantsAlongLength * noOfPlantsAlongBreadth

print("Number of plants along breadth: ",
      int(noOfPlantsAlongBreadth))

print("Number of plants along length: ", int(noOfPlantsAlongLength))
print("Total number of plants: "+ str(totalNoOfPlants))
```

Output

```
deadmercury@DESKTOP-QUKDS0U MINGW64 /e/LEARNING/vit_mca/ITA6017_PYTHON (main)
$ C:/Python311/python.exe e:/LEARNING/vit_mca/ITA6017_PYTHON/DA/DA_1/Q3.py
Enter length of farm:
9
Enter breadth of farm:
4
Number of plants along breadth: 3
Number of plants along length: 5
Total number of plants: 15
```

Question 4

A finance consultant decides to create an application for the public who wants to decide on the type of investment they can make. He needs an application, which when given with the initial amount, no of years of investment 'n' and the expected amount after 'n' years, should tell the rate of interest based on which the customers will choose their investments. Assume that the application works with only simple interest calculation. Can you code for Raman? Use Rate of interest = $((\text{expected amount} - \text{initial amount}) / (\text{initial amount} * \text{no of years})) * 100$.

Input Format:

The first input value read is the initial amount the customer has.

The second input value read is the no of years he want to invest.

The third input value is the expected amount after 'n' years.

Output Format:

Print rate of interest

Code

```
print("Enter the initial amount: ")
initialAmount = float(input())

print("Enter number of years to invest: ")
noOfYears = int(input())

print("Enter expected amount: ")
expectedAmount = float(input())

rateOfInterest = ((expectedAmount - initialAmount) / (initialAmount
* noOfYears)) * 100
```

```
print("Required rate of interest: " + str(rateOfInterest))
```

Output

```
deadmercury@DESKTOP-QUKDS0U MINGW64 /e/LEARNING/vit_mca/ITA6017_PYTHON (main)
$ C:/Python311/python.exe e:/LEARNING/vit_mca/ITA6017_PYTHON/DA/DA_1/Q4.py
Enter the initial amount:
5000
Enter number of years to invest:
2
Enter expected amount:
10000
Required rate of interest: 50.0
```

Question 5

Consider the thirsty crow story where a thirsty crow identifies a jug with little water. It puts pebbles into the water to raise the level of water and drinks it. Assume that the initial reading of the jug is 'm1' ml and the crow can drink water if the level of water has come to 'm2' ml. There are two categories of pebbles small and big in the field. Small pebble can raise the level of water by 'x' ml and big pebble can raise the level of water by 'y' ml. There are 'n' small pebbles. Crow prefers to place small pebbles in jug and then only takes big pebbles. Write an algorithm and the Python code to determine the number of pebbles required to raise the water to 'm2' level. For example, if value of 'm1', 'm2','x','y' and 'n' are 54, 200, 10, 20, 10 respectively then the number of pebbles required is 13.

Input Format:

Read the initial level of water in jug (in ml)

Read the level of water in jug required for drinking (in ml)

Read the height which small pebble will increase (in ml)

Read the height which big pebble will increase (in ml)

Read the number of small pebbles

Output Format:

Number of big pebbles required

Code

```
print("Enter initial water level: ")
initialWaterLevel = int(input())

print("Enter required water level: ")
requiredWaterLevel = int(input())
```

```
print("Height small pebble increases: ")
smallPebbleHeight = int(input())

print("Height big pebble increases: ")
bigPebbleHeight = int(input())

print("Number of small pebbles: ")
smallPebbleCount = int(input())

noOfSmallPebbles = 0

while initialWaterLevel <= requiredWaterLevel and noOfSmallPebbles <
smallPebbleCount:
    initialWaterLevel += smallPebbleHeight
    noOfSmallPebbles += 1

noOfBigPebbles = 0

while initialWaterLevel <= requiredWaterLevel:
    initialWaterLevel += bigPebbleHeight
    noOfBigPebbles += 1

totalPebbles = noOfSmallPebbles + noOfBigPebbles

print("Number of small pebbles required: " + str(noOfSmallPebbles))
print("Number of big pebbles required: " + str(noOfBigPebbles))
```

```
print("Total number of pebbles required " + str(totalPebbles))
```

Output

```
deadmercury@DESKTOP-QUKDS0U MINGW64 /e/LEARNING/vit_mca/ITA6017_PYTHON (main)
$ C:/Python311/python.exe e:/LEARNING/vit_mca/ITA6017_PYTHON/DA/DA_1/Q5.py
Enter initial water level:
54
Enter required water level:
200
Height small pebble increases:
10
Height big pebble increases:
20
Number of small pebbles:
10
Number of small pebbles required: 10
Number of big pebbles required: 3
Total number of pebbles required 13
```

Question 6

Given 'n' points, design a flowchart or an algorithm and write the Python code to determine the pair of points that are far from each other. Distance between two points (x_1, y_1) and (x_2, y_2) is determined using the formula $\sqrt{(x_1 - x_2)^2 + (y_1 - y_2)^2}$. Write a function to determine distance between the two points. Consider only two decimal places of distance for comparison.

Code

```
import math

def distBetPoints(p1, p2):
    x1 = round(p1[0], 2)
    x2 = round(p2[0], 2)
    y1 = round(p1[1], 2)
    y2 = round(p2[1], 2)
    return round(math.sqrt(math.pow((x1 - x2), 2) + math.pow((y1 - y2), 2)), 2)

print("Enter number of points: ")
noOfPoints = int(input())

if noOfPoints < 2:
    print("Enter at least two points for comparison")
    exit()

points = []
```



```

print("\nEnter points: ")
for i in range(noOfPoints):
    print("\nPoint " + str(i + 1) + ": ")
    print("x: ")
    x = float(input())
    print("y: ")
    y = float(input())
    pointTuple = (x, y)
    points.append(pointTuple)

distanceMap = {}

for i in range(noOfPoints - 1):
    for j in range(i + 1, noOfPoints):
        distanceMap[points[i], points[j]] = distBetPoints(points[i],
points[j])

maxDistance = float("-inf")
maxDistancePoint = points[0]
for points, distance in distanceMap.items():
    if distance > maxDistance:
        maxDistance = distance
        maxDistancePoint = points

print(
    "Max distance is between points "

```

```
+ str(maxDistancePoint[0])  
+ " and "  
+ str(maxDistancePoint[1])  
+ " having distance "  
+ str(maxDistance)  
)
```

Output

```
deadmercury@DESKTOP-QUKDS0U MINGW64 /e/LEARNING/vit_mca/ITA6017_PYTHON (main)  
$ C:/Python311/python.exe e:/LEARNING/vit_mca/ITA6017_PYTHON/DA/DA_1/Q6.py  
Enter number of points:  
3  
  
Enter points:  
  
Point 1:  
x:  
2  
y:  
1  
  
Point 2:  
x:  
2  
y:  
3  
  
Point 3:  
x:  
4  
y:  
6  
Max distance is between points (2.0, 1.0) and (4.0, 6.0) having distance 5.39
```

```
deadmercury@DESKTOP-QUKDS0U MINGW64 /e/LEARNING/vit_mca/ITA6017_PYTHON (main)  
$ C:/Python311/python.exe e:/LEARNING/vit_mca/ITA6017_PYTHON/DA/DA_1/Q6.py  
Enter number of points:  
1  
Enter at least two points for comparison
```

Question 7

A person wants to know his Body Mass Index (BMI). He knows his weight in pounds and height in inches. The evaluator knows the formula for calculating BMI using the formula,

$$\text{BMI} = (\text{weight in kilograms}) / (\text{height in m} * \text{height in m})$$

Help the person in finding his BMI by writing a program for him. (Use the conversion formulae: 1 pound = 0.4536 kilograms, 1 inch = 2.54 cms)

Input Format:

Weight of person in pounds.

Height of the person in inches.

Output Format:

BMI of the person calculated using the formula,

$$(\text{weight in kilograms}) / (\text{height in m} * \text{height in m})$$

Code

```
import math

print("Enter weight in pounds: ")
weightInPounds = float(input())

print("Enter height in inches: ")
heightInInches = float(input())

weightInKilos = weightInPounds * 0.4536
heightInMetres = (heightInInches * 2.54) / 100
```

```
bmi = round(weightInKilos / math.pow(heightInMetres, 2), 2)

print("BMI of the person is: " + str(bmi))
```

Output

```
deadmercury@DESKTOP-QUKDS0U MINGW64 /e/LEARNING/vit_mca/ITA6017_PYTHON (main)
$ C:/Python311/python.exe e:/LEARNING/vit_mca/ITA6017_PYTHON/DA/DA_1/Q7.py
Enter weight in pounds:
200
Enter height in inches:
71
BMI of the person is: 27.89

deadmercury@DESKTOP-QUKDS0U MINGW64 /e/LEARNING/vit_mca/ITA6017_PYTHON (main)
$ C:/Python311/python.exe e:/LEARNING/vit_mca/ITA6017_PYTHON/DA/DA_1/Q7.py
Enter weight in pounds:
300
Enter height in inches:
66
BMI of the person is: 48.42
```