Abhinav Srivatsa

21BDS0340

Computer Science Lab

# FAT Exam

**Question 1:**

Raj has n pens. If Raj keeps percentage p of the pens, the remaining will be shared among his friends. Write a Python program to read the total number of pens, say it n. Also read the percentage (p) of pens that Raj will keep with him. Create a function[named share\_YourRegistrationNumber] to Calculate the number of remaining pens (r) that will be shared among his friends.

If no pen is left for sharing display “No pens left to share”.

If percentage of pens fractional part is >=0.5 then raise the value to its higher integer and if it is <0.5 then lower the value to its lower integer.

Input:

First line contains the value of n

Second line contains the value of p

Output:

The value of remaining pens(r)

Constraints:

0≤p≤100 Otherwise print “Percentage should be 0 to 100”

n>=0 Otherwise print “Pens should be >= 0”

**Code:**

def share\_21BDS0340(total\_p, p\_share):

    r = total\_p - round(total\_p \*  p\_share / 100)

    if r == 0:

        r = 'No pens left to share'

    return r

n = int(input())

if n < 0:

    print('Pens should be >= 0')

else:

    p = int(input())

    if p < 0 or p > 100:

        print('Percentage should be 0 to 100')

    else:

        print(share\_21BDS0340(n, p))

**Algorithm:**

share\_21BDS0340(N, P):

Calculate R as N – rounded N \* P / 100

If R is 0, display ‘No pens left to share’

Return R

Read N

If N < 0, then display ‘Pens should be >= 0’

Else

Read P

If P < 0 or P > 100, then display ‘Percentage should be 0 to 100’

Else, display share\_21BDS0340(N, P)

**Output:**

Application, table

Description automatically generated

**Question 2:**

Write a function anaconda\_code(your\_input) to convert the given input to word “Anaconda”. The function accepts a single parameter. The parameter value is to be converted to bits i.e., 8 bits. The bits should be mapped to “Anaconda” which is also 8 characters length. When there is ‘1’, the character should be of capital-case letter; if ‘0’ then small-case letter.

Input:

A single line contains the value of n

Output:

A single line contains ‘Anaconda’ code

Constraint:

If the input is a number, then 0≤input\_number≤255 otherwise print “Number should be ≤255”

**Code:**

def translator\_21BDS0340(n):

    word = 'anaconda'

    letters = list(word)

    bin\_arr = []

    try:

        n = int(n)

        n = str(bin(n))[2:]

        bin\_arr = [n]

    except:

        n = n

        bin\_arr = [format(ord(i), 'b') for i in n]

    output = []

    for i in bin\_arr:

        index = 7

        for j in range(len(i) - 1, -1, -1):

            if i[j] == '1':

                letters[index] = letters[index].upper()

            index -= 1

        output.append(''.join(letters))

        letters = list(word)

    return output

word = input()

print(' '.join(translator\_21BDS0340(word)))

**Algorithm:**

translator\_21BDS0340(N):

Initialise Word as ‘anaconda’

Initialize Letters as a list of Word

Initialise Bin\_Arr as list

Try casting N as integer

If N < 0 or N > 255, then return 'Number should be ≤255'

Calculate N as string of binary of N but slicing the first 2 characters

Except initialise Bin\_Arr as binary version of the characters of N

Initialise Output as list

Loop through Bin\_Arr as I

Initialise Index as 7

Loop through length of I backwards with J

If I of J is 1, then make Letters of Index upper case

Decrement Index by 1

Append Letters to Output

Initialize Letters as a list of Word

Return Output

Read Word

Initialise Output as translator\_21BDS0340(Word)

If Output’s type is list, then display Output joined by spaces

Else, display Output

**Output:**

Table

Description automatically generated