Problem Statement

A doctor has a clinic where he serves his patients. The doctor’s consultation fees are different for different groups of patients depending on their age. If the patient’s age is below 17, fees is 200 INR. If the patient’s age is between 17 and 40, fees is 400 INR. If patient’s age is above 40, fees is 300 INR. Write a code to calculate earnings in a day for which one array/List of values representing age of patients visited on that day is passed as input.

Note:

Age should not be zero or less than zero or above 120

Doctor consults a maximum of 20 patients a day

Enter age value (press Enter without a value to stop):

Example 1:

Input

20

30

40

50

2

3

14

Output

Total Income 2000 INR

Note: Input and Output Format should be same as given in the above example.

For any wrong input display INVALID INPUT

Output Format

Total Income 2000 INR

def total\_amount(ages):

    amount = 0

    for age in ages:

        if age < 17:

            amount += 200

        elif 17 <= age < 40:

            amount += 400

        else:

            amount += 300

    return amount

ages = []

while True:

    i = input("Enter the age : ")

    if i == "":

        print("Invalid input")

        break

    age = int(i)

    if age <= 0 or age > 120:

        print("Invalid input")

        continue

    if len(ages) < 20:

        ages.append(age)

    else:

        print("Exceed the limit")

        break

print("Total Amount:", total\_amount(ages))

2.There is a JAR full of candies for sale at a mall counter. JAR has the capacity N, that is JAR can contain maximum N candies when JAR is full. At any point of time. JAR can have M number of Candies where M<=N. Candies are served to the customers. JAR is never remain empty as when last k candies are left. JAR if refilled with new candies in such a way that JAR get full.

Write a code to implement above scenario. Display JAR at counter with available number of candies. Input should be the number of candies one customer can order at point of time. Update the JAR after each purchase and display JAR at Counter.

Output should give number of Candies sold and updated number of Candies in JAR.

If Input is more than candies in JAR, return: “INVALID INPUT”

Given,

N=10, where N is NUMBER OF CANDIES AVAILABLE

K =< 5, where k is number of minimum candies that must be inside JAR ever.

Example 1:(N = 10, k =< 5)

Input Value

3

Output Value

NUMBER OF CANDIES SOLD : 3

NUMBER OF CANDIES LEFT : 7

Example : (N=10, k<=5)

Input Value

0

Output Value

INVALID INPUT

NUMBER OF CANDIES LEFT : 10

n=int(input("enter the maximum capacity of jar"))

k=int(input("Enter the minimum candles should be in the jar"))

i=int(input("enter the number of candles:"))

if i == 0:

    print("invalid input")

elif(i<=k):

    print("NUMBER OF CANDIES SOLD : ",i)

    print("NUMBER OF CANDIES LEFT :",n-i)

elif i>k:

    print("No sufficient candles")

    print("NUMBER OF CANDIES LEFT :",n)

3.Problem Statement

A washing machine works on the principle of Fuzzy System, the weight of clothes put inside it for washing is uncertain But based on weight measured by sensors, it decides time and water level which can be changed by menus given on the machine control area.

For low level water, the time estimate is 25 minutes, where approximately weight is between 2000 grams or any nonzero positive number below that.

For medium level water, the time estimate is 35 minutes, where approximately weight is between 2001 grams and 4000 grams.

For high level water, the time estimate is 45 minutes, where approximately weight is above 4000 grams.

Assume the capacity of machine is maximum 7000 grams

Where approximately weight is zero, time estimate is 0 minutes.

Write a function which takes a numeric weight in the range [0,7000] as input and produces estimated time as output is: “OVERLOADED”, and for all other inputs, the output statement is

“INVALID INPUT”.

Input should be in the form of integer value –

Output must have the following format –

Time Estimated: Minutes

Example:

Input value

2000

Output value

Time Estimated: 25 minutes

def total\_min(weight):

    min = 0

    if weight ==0:

        print("invalid input")

    elif weight>0 and weight<=2000:

        min=25

    elif weight >=2001 and weight <=4000:

        min= 35

    elif weight >4000 and weight<=7000:

        min=45

    else:

        print("overloaded")

    return (min)

weight=int(input("enter the weightof the clothes:"))

print(total\_min(weight),"minutes")

4.Airport security officials have confiscated several item of the passengers at the security check point. All the items have been dumped into a huge box (array). Each item possesses a certain amount of risk[0,1,2]. Here, the risk severity of the items represent an array[] of N number of integer values. The task here is to sort the items based on their levels of risk in the array. The risk values range from 0 to 2.

Example :

Input :

7 -> Value of N

[1,0,2,0,1,0,2]-> Element of arr[0] to arr[N-1], while input each element is separated by new line.

Output :

0 0 0 1 1 2 2 -> Element after sorting based on risk severity

Example 2:

inpt : 10 -> Value of N

[2,1,0,2,1,0,0,1,2,0] -> Element of arr[0] to arr[N-1], while input each element is separated by a new line.

Output :

0 0 0 0 1 1 1 2 2 2 ->Elements after sorting based on risk severity.

n=int(input("enter the value of N"))

arr=[]

print("enter the items")

for i in range(n):

    item=int(input())

    arr.append(item)

print(sorted(arr))

5. supermarket maintains a pricing format for all its products. A value N is printed on each product. When the scanner reads the value N on the item, the product of all the digits in the value N is the price of the item. The task here is to design the software such that given the code of any item N the product (multiplication) of all the digits of value should be computed(price).

Example 1:

Input :

5244 -> Value of N

Output :

160 -> Price

Explanation:

From the input above

Product of the digits 5,2,4,4

5\*2\*4\*4= 160

Hence, output is 160.

def price(n):

    product=1

    while n!=0:

        rem=n%10

        product=product\*rem

        n=n//10

    return product

n=int(input('Enter the code'))

print(price(n))

1. Problem Statement

An intelligence agency has received reports about some threats. The reports consist of numbers in a mysterious method. There is a number “N” and another number “R”. Those numbers are studied thoroughly and it is concluded that all digits of the number ‘N’ are summed up and this action is performed ‘R’ number of times. The resultant is also a single digit that is yet to be deciphered. The task here is to find the single-digit sum of the given number ‘N’ by repeating the action ‘R’ number of times.

If the value of ‘R’ is 0, print the output as ‘0’.

Example 1:

Input :

99 -> Value of N

3 -> Value of R

Output :

9 -> Possible ways to fill the cistern.

Example 2:

Input :

1234 -> Value of N

2 -> Value of R

Output :

2 -> Possible ways to fill the cistern

1. Marsexploration

Letters in some of the SOS messages are altered by cosmic radiation during transmission. Given the signal received by Earth as a string, , determine how many letters of the SOS message have been changed by radiation.

Example

The original message was SOSSOS. Two of the message's characters were changed in transit.

Function Description

Complete the marsExploration function in the editor below.

marsExploration has the following parameter(s):

string s: the string as received on Earth

Returns

int: the number of letters changed during transmission

Input Format

There is one line of input: a single string, .

Constraints

will contain only uppercase English letters, ascii[A-Z].

Sample Input 0

SOSSPSSQSSOR

Sample Output 0

3

Explanation 0

= SOSSPSSQSSOR, and signal length . They sent SOS messages (i.e.: ).

Expected signal: SOSSOSSOSSOS

Recieved signal: SOSSPSSQSSOR

def marsExploration(s):

    if len(s)% 3!=0:

        return -1

    else:

        count=0

        for i in range(0,len(s),3):

            if s[i]!='S':

                count+=1

            if s[i+1]!='O':

                count+=1

            if s[i+2]!='S':

                count+=1

        return count

s=str(input("enter the string message")).upper()

result=marsExploration(s)

print(result)

1. **Jumpingonccloud**

A child is playing a cloud hopping game. In this game, there are sequentially numbered clouds that can be thunderheads or cumulus clouds. The character must jump from cloud to cloud until it reaches the start again.There is an array of clouds, and an energy level . The character starts from and uses unit of energy to make a jump of size to cloud . If it lands on a thundercloud, , its energy () decreases by additional units. The game ends when the character lands back on cloud .Given the values of , , and the configuration of the clouds as an array , determine the final value of after the game ends.

Example.

The indices of the path are . The energy level reduces by for each jump to . The character landed on one thunderhead at an additional cost of energy units. The final energy level is .

Note: Recall that refers to the modulo operation. In this case, it serves to make the route circular. If the character is at and jumps , it will arrive at .

Function Description

Complete the jumpingOnClouds function in the editor below.

jumpingOnClouds has the following parameter(s):

int c[n]: the cloud types along the path

int k: the length of one jump

Returnsint: the energy level remaining.

Input Format

The first line contains two space-separated integers, and , the number of clouds and the jump distance.The second line contains space-separated integers where . Each cloud is described as follows:

If , then cloud is a cumulus cloud.

If , then cloud is a thunderhead.

Constraints

Sample Input

STDIN Function

----- --------

8 2 n = 8, k = 2

0 0 1 0 0 1 1 0 c = [0, 0, 1, 0, 0, 1, 1, 0]

Sample Output

92