**TCS Coding Questions**

**1.Problem Statement –**

A chocolate factory is packing chocolates into the packets. The chocolate packets here represent an array of N number of integer values. The task is to find the empty packets(0) of chocolate and push it to the end of the conveyor belt(array).

**Example 1:**

N=7 and arr = [4,5,0,1.9,0,5,0].

There are 3 empty packets in the given set. These 3 empty packets represented as O should be pushed towards the end of the array

**Input :**

7  – Value of N

[4,5,0,1,0,0,5] – Element of arr[O] to arr[N-1],While input each element is separated by newline

**Output:**

4 5 1 9 5 0 0 0

**Example 2:**

**Input:**

6 — Value of N.

[6,0,1,8,0,2] – Element of arr[0] to arr[N-1], While input each element is separated by newline

**Output:**

6 1 8 2 0 0

Code:

n=int(input())  
j=0  
L=[0 for i in range(n)]  
for i in range(n):  
    a=int(input())  
    if a!=0:  
        L[j]=a  
        j+=1  
for i in L:  
    print(i,end=" ")

**2 Problem Statement –**

Joseph is learning digital logic subject which will be for his next semester. He usually tries to solve unit assignment problems before the lecture. Today he got one tricky question. The problem statement is “A positive integer has been given as an input. Convert decimal value to binary representation. Toggle all bits of it after the most significant bit including the most significant bit. Print the positive integer value after toggling all bits”.

**Constrains-**

1<=N<=100

**Example 1:**

**Input :**

10  -> Integer

**Output :**

5    -> result- Integer

**Explanation:**

Binary representation of 10 is 1010. After toggling the bits(1010), will get 0101 which represents “5”. Hence output will print “5”.

Code:

import math  
n=int(input())  
k=(1<<int(math.log2(n))+1)-1  
print(n^k)

**3.Problem Statement –**

Jack is always excited about sunday. It is favourite day, when he gets to play all day. And goes to cycling with his friends. so every time when the months starts he counts the number of sundays he will get to enjoy. Considering the month can start with any day, be it Sunday, Monday…. Or so on.

Count the number of Sunday jack will get within n number of days.

**Example 1:**

**Input**

mon-> input String denoting the start of the month.

13  -> input integer denoting the number of days from the start of the month.

**Output :**

2    -> number of days within 13 days.

**Explanation**:

The month start with mon(Monday). So the upcoming sunday will arrive in next 6 days. And then next Sunday in next 7 days and so on. Now total number of days are 13. It means 6 days to first sunday and then remaining 7 days will end up in another sunday. Total 2 sundays may fall within 13 days.

Code:

from collections import defautdict  
m=defaultdict(int)  
m["mon"]=6  
m["tue"]=5  
m["wed"]=4  
m["thu"]=3  
m["fri"]=2  
m["sat"]=1  
m["sun"]=0  
s=input()  
a=int(input())  
ans=0  
if a-m[s[0:3]]>=1:  
  ans=1+(a-m[s[0:3]])//7  
print(ans)

**4. Problem Statement –**

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:**

input : 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.

**Explanation:**

In the above example, the input is an array of size N consisting of only 0’s, 1’s and 2s. The output is a sorted array from 0 to 2 based on risk severity.

Code:

n=int(input())  
a=list(map(int,input().split()))  
l=0  
m=0  
h=n-1  
while m<=h:  
  if a[m]==0:  
    a[l],a[m]=a[m],a[l]  
    l+=1  
    m+=1  
  elif a[m]==1:  
    m+=1  
  else:  
    a[m],a[h]=a[h],a[m]  
    h-=1  
for i in a:  
  print(i,end=" ")

**5. Problem Statement –**

Given an integer array Arr of size N the task is to find the count of elements whose value is greater than all of its prior elements.

Note : 1st element of the array should be considered in the count of the result.

For example,

Arr[]={7,4,8,2,9}

As 7 is the first element, it will consider in the result.

8 and 9 are also the elements that are greater than all of its previous elements.

Since total of  3 elements is present in the array that meets the condition.

Hence the output = 3.

**Example 1:**

**Input**

5 -> Value of N, represents size of Arr

7-> Value of Arr[0]

4 -> Value of Arr[1]

8-> Value of Arr[2]

2-> Value of Arr[3]

9-> Value of Arr[4]

**Output :**

3

**Example 2:**

5   -> Value of N, represents size of Arr

3  -> Value of Arr[0]

4 -> Value of Arr[1]

5 -> Value of Arr[2]

8 -> Value of Arr[3]

9 -> Value of Arr[4]

**Output :**

5

**Constraints**

1<=N<=20

1<=Arr[i]<=10000

Code:

import sys  
n=int(input())  
c=0  
m=-sys.maxsize-1  
while n:  
    n-=1  
    a=int(input())  
    if a>m:  
        m=a  
        c+=1  
print(c)

**6.Problem Statement –**

A 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.

Code:

n=input()  
p=1  
for i in n:  
    p\*=int(i)  
print(p)

**7.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.

**Explanation:**

Here, the number N=99

1. Sum of the digits N: 9+9 = 18
2. Repeat step 2 ‘R’ times i.e. 3 tims  (9+9)+(9+9)+(9+9) = 18+18+18 =54
3. Add digits of 54 as we need a single digit 5+4

Hence , the output is 9.

**Example 2:**

**Input :**

1234   -> Value of N

2      -> Value of R

**Output :**

2  -> Possible ways to fill the cistern

**Explanation:**

Here, the number N=1234

1. Sum of the digits of N: 1+2+3+4 =10
2. Repeat step 2 ‘R’ times i.e. 2 times  (1+2+3+4)+(1+2+3+4)= 10+10=20
3. Add digits of 20 as we need a single digit. 2+0=2

Hence, the output is 2.

**Constraints:**

0<N<=1000

0<=R<=50

The Input format for testing

The candidate has to write the code to accept 2 input(s)

First input- Accept value for N (positive integer number)

Second input: Accept value for R(Positive integer number)

The output format for testing

The output should be a positive integer number or print the message (if any) given in the problem statement. (Check the output in Example 1, Example 2).

Code:

s=input()  
n=int(input())  
sum=0  
for i in s:  
    sum+=int(i)  
sum\*=n  
s=str(sum)  
while len(s)>1:  
    sum=0  
    for i in s:  
        sum+=int(i)  
    s=str(sum)  
  
print(s)

**8.Problem Statement**

Given a string S(input consisting) of ‘\*’ and ‘#’. The length of the string is variable. The task is to find the minimum number of ‘\*’ or ‘#’ to make it a valid string. The string is considered valid if the number of ‘\*’ and ‘#’ are equal. The ‘\*’ and ‘#’ can be at any position in the string.

Note : The output will be a positive or negative integer based on number of ‘\*’ and ‘#’ in the input string.

* (\*>#): positive integer
* (#>\*): negative integer
* (#=\*): 0

**Example 1:**

**Input 1:**

* ###\*\*\*   -> Value of S

**Output :**

0   → number of \* and # are equal

Code:

s="Hello"

a=0

b=0

for i in s:

if i=='\*':

a+=1

elif i=='#':

b+=1

print(a-b)

**9.Problem Statement**

Given an integer array Arr of size N the task is to find the count of elements whose value is greater than all of its prior elements.

**Note :** 1st element of the array should be considered in the count of the result.

**For example,**

Arr[]={7,4,8,2,9}

As 7 is the first element, it will consider in the result.

8 and 9 are also the elements that are greater than all of its previous elements.

Since total of  3 elements is present in the array that meets the condition.

Hence the output = 3.

**Example 1:**

**Input**

5 -> Value of N, represents size of Arr

7-> Value of Arr[0]

4 -> Value of Arr[1]

8-> Value of Arr[2]

2-> Value of Arr[3]

9-> Value of Arr[4]

**Output :**

3

**Example 2:**

5   -> Value of N, represents size of Arr

3  -> Value of Arr[0]

4 -> Value of Arr[1]

5 -> Value of Arr[2]

8 -> Value of Arr[3]

9 -> Value of Arr[4]

**Output :**

5

**Constraints**

1<=N<=20

1<=Arr[i]<=10000

Code:

import sys

n=int(input())

c=0

m=-sys.maxsize-1

while n:

n-=1

a=int(input())

if a>m:

m=a

c+=1

print(c)

**10.Problem Statement**

A parking lot in a mall has RxC number of parking spaces. Each parking space will either be  empty(0) or full(1). The status (0/1) of a parking space is represented as the element of the matrix. The task is to find index of the prpeinzta row(R) in the parking lot that has the most of the parking spaces full(1).

**Note :**

RxC- Size of the matrix

Elements of the matrix M should be only 0 or 1.

**Example 1:**

**Input :**

3   -> Value of R(row)

3    -> value of C(column)

[0 1 0 1 1 0 1 1 1] -> Elements of the array M[R][C] where each element is separated by new line.

**Output :**

3  -> Row 3 has maximum number of 1’s

**Example 2:**

**input :**

4 -> Value of R(row)

3 -> Value of C(column)

[0 1 0 1 1 0 1 0 1 1 1 1] -> Elements of the array M[R][C]

**Output :**

4  -> Row 4 has maximum number of 1’s

Code:

r=int(input())

c=int(input())

sum=0

m=0

id=0

for i in range(r):

for j in range(c):

sum+=int(input())

if sum>m:

m=sum

id=i+1

sum=0

print(id)

**11.Problem Statement**

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 AVAILABLE : 7

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

* **Input Value**
  + 0
* **Output Value**
  + INVALID INPUT
  + NUMBER OF CANDIES LEFT : 10

Code:

total\_candies = 10

no\_of\_candies = int(input())

if no\_of\_candies in range(1, 6):

print('No. of Candies Sold:',no\_of\_candies)

print('No. of Candies Left:',total\_candies-no\_of\_candies)

else:

print("Invalid Input")

print('No. of Candies Left:',total\_candies)

**14.Problem statement**

Selection of MPCS exams include a fitness test which is conducted on ground. There will be a batch of 3 trainees, appearing for running test in track for 3 rounds. You need to record their oxygen level after every round. After trainee are finished with all rounds, calculate for each trainee his average oxygen level over the 3 rounds and select one with highest oxygen level as the most fit trainee. If more than one trainee attains the same highest average level, they all need to be selected.

**Display the most fit trainee (or trainees) and the highest average oxygen level.**

**Note:**

* **The oxygen value entered should not be accepted if it is not in the range between 1 and 100.**
* If the calculated maximum average oxygen value of trainees is below 70 then declare the trainees as unfit with meaningful message as “All trainees are unfit.
* Average Oxygen Values should be rounded.

**Example 1:**

* **INPUT VALUES**

            95

            92

            95

            92

            90

            92

            90

            92

            90

* **OUTPUT VALUES**
  + Trainee Number : 1
  + Trainee Number : 3

**Note:**

Input should be 9 integer values representing oxygen levels entered in order as

**Round 1**

* Oxygen value of trainee 1
* Oxygen value of trainee 2
* Oxygen value of trainee 3

**Round 2**

* Oxygen value of trainee 1
* Oxygen value of trainee 2
* Oxygen value of trainee 3

**Round 3**

* Oxygen value of trainee 1
* Oxygen value of trainee 2
* Oxygen value of trainee 3

**Output must be in given format as in above example. For any wrong input final output should display “INVALID INPUT”**

**Code:**

trainee = [[],[],[],[]]

for i in range(3):

for j in range(3):

trainee[i].append(int(input()))

if (trainee[i][-1]) not in range(1,101):

print("invalid input")

for i in range(3):

trainee[3].append((trainee[2][i]+trainee[1][i]+trainee[0][i])//3)

maximum = max(trainee[3])

for i in range(3):

if trainee[3][i] < 70 :

print("Trainee {0} is unfit".format(i+1))

elif trainee[3][i] == maximum:

print("Trainee Number: ",i+1)

**14.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

Code:

n = int(input())

if n==0:

print("Time Estimated : 0 Minutes")

elif n in range(1,2001):

print("Time Estimated : 25 Minutes")

elif n in range(2001,4001):

print("Time Estimated : 35 Minutes")

elif n in range(4001,7001):

print("Time Estimated : 45 Minutes")

else:

print("INVALID INPUT")

**15.Problem statement**

**We want to estimate the cost of painting a property. Interior wall painting cost is Rs.18 per sq.ft. and exterior wall painting cost is Rs.12 per sq.ft.**

**Take input as**  
1. Number of Interior walls  
2. Number of Exterior walls  
3. Surface Area of each Interior 4. Wall in units of square feet  
Surface Area of each Exterior Wall in units of square feet

**If a user enters zero  as the number of walls then skip Surface area values as User may don’t  want to paint that wall.**

**Calculate and display the total cost of painting the property  
Example 1:**

6  
3  
12.3  
15.2  
12.3  
15.2  
12.3  
15.2  
10.10  
10.10  
10.00  
Total estimated Cost : 1847.4 INR

**Code:**

interior\_walls = int(input())  
exterior\_walls = int(input())  
if interior\_walls:  
    int\_walls = []  
    for i in range(interior\_walls):  
        int\_walls.append(float(input()))  
  
if exterior\_walls:  
    ext\_walls = []  
    for i in range(exterior\_walls):  
        ext\_walls.append(float(input()))  
if exterior\_walls < 0 or interior\_walls < 0:  
    print(“Invalid Input”)  
    exit()  
if exterior\_walls and interior\_walls:  
    print("Total estimated Cost : ",(sum(int\_walls)\*18+sum(ext\_walls)\*12),"INR")  
else:  
    if exterior\_walls:  
        print("Total estimated Cost : ",sum(ext\_walls)\*12,"INR")  
    elif interior\_walls:  
         print("Total estimated Cost : ",sum(int\_walls)\*18,"INR")  
    else:  
        print("Total estimated Cost : 0.0 INR")

16.Problem statement:

A City Bus is a Ring Route Bus which runs in circular fashion. That is, Bus once starts at the Source Bus Stop, halts at each Bus Stop in its Route and at the end it reaches the Source Bus Stop again.  
If there are n number of Stops and if the bus starts at Bus Stop 1, then after nth Bus Stop, the next stop in the Route will be Bus Stop number 1 always.  
If there are n stops, there will be n paths. One path connects two stops. Distances (in meters) for all paths in Ring Route is given in array Path [] as given below:  
Path = [800, 600, 750, 900, 1400, 1200, 1100, 1500]  
Fare is determined based on the distance covered from source to destination stop as  Distance between Input Source and Destination Stops can be measured by looking at values in array Path[] and fare can be calculated as per following criteria:

* If d =1000 metres, then fare=5 INR
* (When calculating fare for others, the calculated fare containing any fraction value should be ceiled. For example, for distance 900n when fare initially calculated is 4.5 which must be ceiled to 5)

Path is circular in function. Value at each index indicates distance till current stop from the previous one. And each index position can be mapped with values at same index in BusStops [] array, which is a string array holding abbreviation of names for all stops as-  
**“THANERAILWAYSTN” = ”TH”, “GAONDEVI” = “GA”, “ICEFACTROY” = “IC”, “HARINIWASCIRCLE” = “HA”, “TEENHATHNAKA” = “TE”, “LUISWADI” = “LU”, “NITINCOMPANYJUNCTION” = “NI”, “CADBURRYJUNCTION” = “CA”**

Given, n=8, where n is number of total BusStops.  
**BusStops = [ “TH”, ”GA”, ”IC”, ”HA”, ”TE”, ”LU”, ”NI”,”CA” ]**

Write a code with function getFare(String Source, String Destination) which take Input as source and destination stops(in the format containing first two characters of the Name of the Bus Stop) and calculate and return travel fare.

Example 1:  
**Input Values**  
ca  
Ca

**Output Values**  
INVALID OUTPUT

Example 2:  
**Input Values**NI  
HA  
**Output Values**23.0 INR

Note: Input and Output should be in format given in example.  
Input should not be case sensitive and output should be in the format   INR

Code:

import math  
def getFare(source,destination):  
    route=[ [ "TH", "GA", "IC", "HA", "TE", "LU", "NI", "CA"],  
    [800,600,750,900,1400,1200,1100,1500]  
        ]  
    fare = 0.0  
    if not (source in route[0] and destination in route[0]):  
        print("Invalid Input")  
        exit()  
    if route[0].index(source) < route[0].index(destination):  
        for i in range(route[0].index(source),route[0].index(destination)+1):  
            fare+=route[1][i]  
    elif route[0].index(destination) < route[0].index(source):  
        for i in range(route[0].index(source)+1,len(route[0])):  
            fare+=route[1][i]  
        for i in range(0,route[0].index(destination)+1):  
            fare+=route[1][i]  
    return float(math.ceil(fare\*0.005))  
     
source = input()  
destination = input()  
fare = getFare(source,destination)  
if fare == 0:  
    print("Invalid Input")  
else:  
    print(fare)

17.Problem Statement:

Chain Marketing Organization has has a scheme for income generation, through which its members generate income for themselves. The scheme is such that suppose A joins the scheme and makes R and V to join this scheme then A is Parent Member of R and V who are child Members. When any member joins the scheme then the parent gets total commission of 10% from each of its child members.  
Child members receive commission of 5% respectively. If a Parent member does not have any member joined under him, then he gets commission of 5%.  
Take name of the members joining the scheme as input.  
Display how many members joined the scheme including parent member.Calculate the Total commission gained by each members in the scheme. The fixed amount for joining the scheme is Rs.5000 on which commission will be generated  
**SchemeAmount = 5000**

**Example 1: When there are more than one child members   
Input : (Do not give input prompts.Accept values as follows. )**Amit                     //Enter parent Member as this  
Y                           //Enter Y if  Parent member has child members otherwise enter N  
Rajesh,Virat        //Enter names of child members of Amit in comma separated  
**Output:(Final Output must be in format given below.**)  
TOTAL MEMBERS:3  
COMISSION DETAILS  
Amit: 1000 INR  
Rajesh :250 INR  
Virat: 250 INR

**Example 2: When there is only one child member in the hierarchy  
Input :**Amit  
Y  
Rajesh  
**Output:**Total Members: 2   
Comission Details  
Amit: 500 INR  
Rajesh: 250 INR

Code:

parent = input()  
Yes\_No = input()  
if Yes\_No == "N":  
    print("TOTAL MEMBERS:1\nCOMMISSION DETAILS\n{0}: 250 INR".format(parent))  
elif Yes\_No == "Y":  
    child=list(map(str,input().split(",")))  
    print("TOTAL MEMBERS:{}".format(len(child)+1))  
    print("COMMISSION DETAILS \n{0}:{1} INR".format(parent,len(child)\*500))  
    for i in child:  
        print("{0}:250 INR".format(i))

18.Problem Statement:

**FULLY AUTOMATIC VENDING MACHINE –**dispenses your cuppa on just press of button. A vending machine can serve range of products as follows:

Coffee

1. Espresso Coffee
2. Cappuccino Coffee
3. Latte Coffee

Tea

1. Plain Tea
2. Assam Tea
3. Ginger Tea
4. Cardamom Tea
5. Masala Tea
6. Lemon Tea
7. Green Tea
8. Organic Darjeeling Tea

Soups

1. Hot and Sour Soup
2. Veg Corn Soup
3. Tomato Soup
4. Spicy Tomato Soup

Beverages

1. Hot Chocolate Drink
2. Badam Drink
3. Badam-Pista Drink

**Write a program to take input for main menu & sub menu and display the name of sub menu selected in the following format (enter the first letter to select main menu):**

**Welcome to CCD**

**Enjoy your**

**Example 1:**

* Input:
  + c
  + 1
* Output
  + Welcome to CCD!
  + Enjoy your Espresso Coffee!

**Example 2:**

* Input
  + t
  + 9
* **Output**
  + INVALID OUTPUT!

Code:

menu = [['Espresso Coffee','Cappuucino Coffee','Latte Coffee'], ['Plain Tea',  
  
'Assam Tea','Ginger Tea','Cardamom Tea','Masala Tea','Lemon Tea','Green Tea',  
  
'Organic Darjeeling Tea'], ['Hot and Sour Soup','Veg Corn Soup','Tomato Soup',  
  
'Spicy Tomato Soup'], ['Hot Chocolate Drink', 'Badam Drink',  
  
'Badam-Pista Drink']]  
  
m = input()  
  
if m=='c' or m=='t' or m=='s' or m=='b':  
  
    if m=='c':  
  
        submenu = int(input())  
  
        if submenu in range(3):  
  
            print('Welcome to CCD!\nEnjoy your {}!'.format(menu[0][submenu-1]))  
  
        else:  
  
            print("INVALID INPUT")  
  
    if m=='t':  
  
        submenu = int(input())  
  
        if submenu in range(8):  
  
            print('Welcome to CCD!\nEnjoy your {}!'.format(menu[1][submenu-1]))  
  
        else:  
  
            print("INVALID INPUT")  
  
    if m=='s':  
  
        submenu = int(input())  
  
        if submenu in range(4):  
  
            print('Welcome to CCD!\nEnjoy your {}!'.format(menu[2][submenu-1]))  
  
        else:  
  
            print("INVALID INPUT")  
  
    if m=='b':  
  
        submenu = int(input())  
  
        if submenu in range(3):  
  
            print('Welcome to CCD!\nEnjoy your {}!'.format(menu[3][submenu-1]))  
  
        else:  
  
            print("INVALID INPUT")  
  
else:  
  
    print("INVALID INPUT!")

19.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 - 500  
  30 - 400  
  40 - 400  
  50 - 300  
  2 -200  
  3 200  
  14 200
* 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 2100 INR

Code:

age = []  
  
for i in range(20):  
  
    m = input()  
  
    if m == "":  
  
        break  
  
    elif int(m) in range(0,120):  
  
        age.append(int(m))  
  
    else:  
  
        print("INVALID INPUT")  
  
        exit()  
  
fees = 0  
  
for i in age:  
  
    if i < 17:  
  
        fees+=200  
  
    elif i <40:  
  
        fees+=400  
  
    else:  
  
        fees+=300  
  
print("Total Income {} INR".format(fees))

20.Problem Statement:

To check whether a year is leap or not

**Step 1:**

* We first divide the year by 4.
* If it is not divisible by 4 then it is not a leap year.
* If it is divisible by 4 leaving remainder 0

**Step 2:**

* We divide the year by 100
* **If it is not divisible by 100 then it is a leap year.**
* If it is divisible by 100 leaving remainder 0

**Step 3:**

* We divide the year by 400
* If it is not divisible by 400 then it is a leap year.
* If it is divisible by 400 leaving remainder 0

**Then it is a leap year**

Code:

num = int(input("Enter the year you want to check if is leap year or not: "))  
  
#take input year from the user to check if it is a leap year  
  
if(num%4 == 0):  
  
 #check if the number is divisible by 4 and if true move to next loop  
  
   if(num%100 == 0):  
  
     #check if the input year is divisible by 100 and if true move to next loop  
  
       if(num%400 == 0):  
  
           print("The year {} is a leap year".format(num))  
  
           #the input year is divisible by 4, 100 and 400, hence leap year.  
  
       else:  
  
           print("The year {} is Not a leap year".format(num))  
  
   else:  
  
       print("The year {} is a leap year".format(num))  
  
       #if the number is divisible by both 4 and 100 it is a leap year  
  
else:  
  
   print("The year {} is Not a leap year".format(num))  
  
   #if the input num is not divisible by 4 then it can not be a leap year altogether.

21.Problem statement

 Write a code to check whether no is prime or not. Condition use function check() to find whether entered no is positive or negative ,if negative then enter the no, And if yes pas no as a parameter to prime() and check whether no is prime or not?

* **Whether the number is positive or not, if it is negative then print the message “please enter the positive number”**
* **It is positive then call the function prime and check whether the take positive number is prime or not.**

Code:

def prime(n):  
  
   if n > 1:  
  
    for i in range(2, n):  
  
        if (n % i) == 0:  
  
            print(n, "is not a prime number")  
  
            break  
  
    else:  
  
        print(n, "is a prime number")  
  
num = int(input("enter a number: "))  
  
if (num > 0):  
  
    prime(num)  
  
else:  
  
    print("please enter a positive number")

22.Problem statement:

Find the 15th term of the series?

0,0,7,6,14,12,21,18, 28

**Explanation :**In this series the odd term is increment of 7 {0, 7, 14, 21, 28, 35 – – – – – – }

                        And even term is a increment of 6 {0, 6, 12, 18, 24, 30 – – – – – – }

Code:

num = int(input('enter the number: '))  
  
a=0  
  
b=0  
  
for i in range(1,num+1):  
  
    if(i%2!=0):  
  
        a= a+7  
  
    else:  
  
        b = b+6  
  
if(num%2!=0):  
  
    print(' {} term of series is {}'.format(num,a-7))  
  
else:  
  
    print('{} term of series is {}'.format(num,b-6))

23.Problem Statement:

Consider the following series: 1, 1, 2, 3, 4, 9, 8, 27, 16, 81, 32, 243, 64, 729, 128, 2187 …

This series is a mixture of 2 series – all the odd terms in this series form a geometric series and all the even terms form yet another geometric series. Write a program to find the Nth term in the series.

The value N in a positive integer that should be read from STDIN. The Nth term that is calculated by the program should be written to STDOUT. Other than value of n th term,no other character / string or message should be written to STDOUT. For example , if N=16, the 16th term in the series is 2187, so only value 2187 should be printed to STDOUT.

You can assume that N will not exceed 30.

Code:

n = int(input('enter the number: '))  
  
a= 1  
  
b= 1  
  
for i in range(1, n+1):  
  
    if(i%2!=0):  
  
         
  
        a = a\*2  
  
    else:        
  
        b = b\*3  
  
if(n%2!=0):  
  
    print('\n{} term of series is {}\t'.format(n,a/2))  
  
else:  
  
    print('\n{} term of series is {}\t'.format(n,a/2))

24.Problem Statement:

Consider the below series :

0, 0, 2, 1, 4, 2, 6, 3, 8, 4, 10, 5, 12, 6, 14, 7, 16, 8

This series is a mixture of 2 series all the odd terms in this series form even numbers in ascending order and every even terms is derived from the previous  term using the formula (x/2)

Write a program to find the nth term in this series.

The value n in a positive integer that should be read from STDIN the nth term that is calculated by the program should be written to STDOUT. Other than the value of the nth term no other characters /strings or message should be written to STDOUT.

For example if n=10,the 10 th term in the series is to be derived from the 9th term in the series. The 9th term is 8 so the 10th term is (8/2)=4. Only the value 4 should be printed to STDOUT.

You can assume that the n will not exceed 20,000.

Code:

n = int(input('enter the number:'))  
  
a=0  
b=0  
  
for i in range(1,n+1):  
 if(i%2!=0):  
 a= a+2  
 else:  
 b= b+1  
  
if(n%2!=0):  
 print('{}'.format(a-2))  
else:  
 print('{}'.format(b-1))