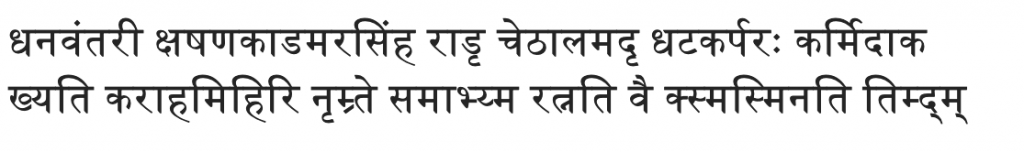
**TCS Coding Questions**

1.Our hoary culture had several great persons since time immemorial and king vikramaditya’s nava ratnas (nine gems) belongs to this ilk.They are named in the following shloka:



Among these, Varahamihira was an astrologer of eminence and his book Brihat Jataak is recokened as the ultimate authority in astrology.

He was once talking with Amarasimha,another gem among the nava ratnas and the author of Sanskrit thesaurus, Amarakosha.

Amarasimha wanted to know the final position of a person, who starts from the origin 0 0 and travels per following scheme.

Scheme

* He first turns and travels 10 units of distance
* His second turn is upward for 20 units
* Third turn is to the left for 30 units
* Fourth turn is the downward for 40 units
* Fifth turn is to the right(again) for 50 units

… And thus he travels, every time increasing the travel distance by 10 units.

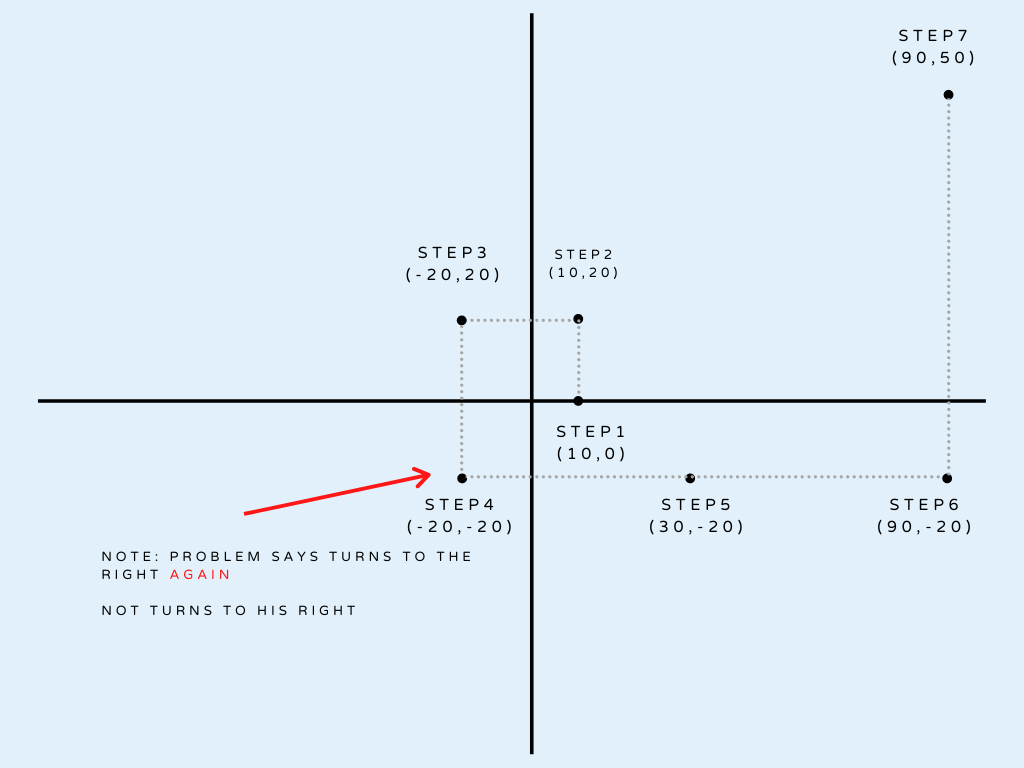
Test Cases

Case 1

* Input : 3
* Expected Output :-20 20

Case 2

* Input : 7
* Expected Output : 90 -20



**2.**Find the nth term of the 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.

**Test Case 1**

* Input- 16
* Expected Output – 2187

**Test Case 2**

* Input- 13
* Expected Output – 64

**3**.Given a maximum of 100 digit numbers as input, find the difference between the sum of odd and even position digits

Test Cases

Case 1

* Input: 4567
* Expected Output: 2

**Explanation :** Odd positions are 4 and 6 as they are pos: 1 and pos: 3, both have sum 10. Similarly, 5 and 7 are at even positions pos: 2 and pos: 4 with sum 12. Thus, difference is 12 – 10 = 2

**4.Consider the below series:**  
**1, 2, 1, 3, 2, 5, 3, 7, 5, 11, 8, 13, 13, 17…..**

This series is a mixture of 2 series. The odd terms in this series form a Fibonacci series and all the even terms are the prime numbers in ascending order. Write a program to find the Nth term in this series

**5.**An automobile company manufactures both a two wheeler (TW) and a four wheeler (FW). A company manager wants to make the production of both types of vehicle according to the given data below:

* 1st data, Total number of vehicle (two-wheeler + four-wheeler)=v
* 2nd data, Total number of wheels = W

The task is to find how many two-wheelers as well as four-wheelers need to manufacture as per the given data.

**Input :**

* 200  -> Value of V
* 540   -> Value of W

**Output :** TW =130 FW=70

**Explanation:**

130+70 = 200 vehicles

(70\*4)+(130\*2)= 540 wheels

**Constraints :**

* 2<=W
* W%2=0
* V<W

 Print “INVALID INPUT” , if inputs did not meet the constraints.

**6**.There is a jar full of candies for sale at a mall counter. The jar has the capacity N, that is JAR can contain maximum N Candies when a JAR is full. At any point in time, JAR can have an M number of candies where M<=N. Candies are served to the customers. JAR is never remaining empty as when the last K candidates are left, JAR is refilled with new candidates in such a way that JAR gets full.

Write the code to implement the above scenario. Display JAR at the counter with the available number of candies.

Input should be the number of candies one customer orders at a point in time. Update the JAR after every purchase and display JAR at the counter. The output should give the number of candies sold and the updated number of candies in the JAR. If the input is more than the number of 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

**7**.The selection of MPCS exams includes a fitness test which is conducted on the ground. There will be a batch of 3 trainees, appearing for a running test on track for 3 rounds.

You need to record their oxygen level after every round. After trainees are finished with all rounds, calculate for each trainee his average oxygen level over the 3 rounds and select the one with the highest average oxygen level as the fittest trainee. If more than one trainee attains the same highest average level, they all need to be selected. Display the fittest trainee(or trainers) 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”**

**8**.A party has been organised on cruise. The party is organised for a limited time(T). The number of guests entering (E[i]) and leaving (L[i]) the party at every hour is represented as elements of the array. The task is to find the maximum number of guests present on the cruise at any given instance within T hours.

**Example 1:**

**Input :**

* 5    -> Value of T
* [7,0,5,1,3]  -> E[], Element of E[0] to E[N-1], where input each element is separated by new line
* [1,2,1,3,4]   -> L[], Element of L[0] to L[N-1], while input each element is separate by new line.

**Output :**

8     -> Maximum number of guests on cruise at an instance.

**Explanation:**

1st hour:

Entry : 7 Exit: 1

No. of guests on ship : 6

2nd hour :

Entry : 0 Exit : 2

No. of guests on ship : 6-2=4

Hour 3:

Entry: 5 Exit: 1

No. of guests on ship : 4+5-1=8

Hour 4:

Entry : 1 Exit : 3

No. of guests on ship : 8+1-3=6

Hour 5:

Entry : 3 Exit: 4

No. of guests on ship: 6+3-4=5

Hence, the maximum number of guests within 5 hours is 8.

**9**.At a fun fair, a street vendor is selling different colours of balloons. He sells N number of different colours of balloons (B[]). The task is to find the colour (odd) of the balloon which is present odd number of times in the bunch of balloons.

**Note:**If there is more than one colour which is odd in number, then the first colour in the array which is present odd number of times is displayed. The colours of the balloons can all be either upper case or lower case in the array. If all the inputs are even in number, display the message “All are even”.

**Example 1:**

* 7  -> Value of N
* [r,g,b,b,g,y,y]  -> B[] Elements B[0] to B[N-1], where each input element is sepārated by ṉew line.

**Output :**

* r -> [r,g,b,b,g,y,y]  -> “r” colour balloon is present odd number of times in the bunch.

**Explanation:**

From the input array above:

* r: 1 balloon
* g: 2 balloons
* b:  2 balloons
* y : 2 balloons

Hence , r is only the balloon which is odd in number.

**10**.The Caesar cipher is a type of substitution cipher in which each alphabet in the plaintext or messages is shifted by a number of places down the alphabet.  
For example,with a shift of 1, P would be replaced by Q, Q would become R, and so on.  
To pass an encrypted message from one person to another, it is first necessary that both parties have the ‘Key’ for the cipher, so that the sender may encrypt and the receiver may decrypt it.  
Key is the number of OFFSET to shift the cipher alphabet. Key can have basic shifts from 1 to 25 positions as there are 26 total alphabets.  
As we are designing custom Caesar Cipher, in addition to alphabets, we are considering numeric digits from 0 to 9. Digits can also be shifted by key places.  
For Example, if a given plain text contains any digit with values 5 and keyy =2, then 5 will be replaced by 7, “-”(minus sign) will remain as it is. Key value less than 0 should result into “INVALID INPUT”

**Example 1:**Enter your PlainText: All the best  
Enter the Key: 1

The encrypted Text is: Bmm uif Cftu

Write a function CustomCaesarCipher(int key, String message) which will accept plaintext and key as input parameters and returns its cipher text as output.