

# TCS

**1.Problem Statement** – 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.

**Example :**

**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 :**

- $2 \leq W$
- $W \% 2 = 0$
- $V < W$

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

**2.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

3. 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**

- $1 \leq N \leq 20$
- $1 \leq \text{Arr}[i] \leq 10000$

4. A parking lot in a mall has  $R \times C$  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 :**

$R \times C$ - 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] -> Elements of the array M[R][C]

**Output :**

4 -> Row 4 has maximum number of 1's

---

5.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.

**Example 2:****Input:**

4 -> Value of T

[3,5,2,0] -> E[], Element of E[0] to E[N-1], where input each element is separated by new line.

[0,2,4,4] -> L[], Element of L[0] to L[N-1], while input each element in separated by new line

**Output:**

6

Cruise at an instance

**Explanation:**

Hour 1:

Entry: 3 Exit: 0

No. of guests on ship: 3

Hour 2:

Entry : 5 Exit : 2

No. of guest on ship:  $3+5-2=6$

Hour 3:

Entry : 2 Exit: 4

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

Hour 4:

Entry: 0 Exit : 4

No. of guests on ship :  $4+0-4=0$

Hence, the maximum number of guests within 5 hours is 6.

The input format for testing

The candidate has to write the code to accept 3 input.

First input- Accept value for number of T(Positive integer number)

Second input- Accept T number of values, where each value is separated by a new line.

Third input- Accept T number of values, where each value is separated by a new line.

The output format for testing

The output should be a positive integer number or a message as given in the problem statement(Check the output in Example 1 and Example 2)

### Constraints:

- $1 \leq T \leq 25$
- $0 \leq E[i] \leq 500$
- $0 \leq L[i] \leq 500$

6. 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 separated by new 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.

### Example 2:

#### Input:

- 10 -> Value of N
- [a,b,b,b,c,c,c,a,f,c] -> B[], elements B[0] to B[N-1] where input each element is separated by new line.

#### Output :

b-> 'b' colour balloon is present odd number of times in the bunch.

#### Explanation:

From the input array above:

- a: 2 balloons
- b: 3 balloons
- c: 4 balloons
- f: 1 balloons

Here, both 'b' and 'f' have odd number of balloons. But 'b' colour balloon occurs first.  
Hence , b is the output.

#### Input Format for testing

The candidate has to write the code to accept: 2 input

- First input: Accept value for number of N(Positive integer number).
- Second Input : Accept N number of character values (B[]), where each value is separated by a new line.

#### Output format for testing

The output should be a single literal (Check the output in example 1 and example 2)

#### Constraints:

- $3 \leq N \leq 50$
- $B[i] = \{a-z\} \text{ or } \{A-Z\}$

7. 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  $M \leq N$ . Candies are served to the customers. JAR is never remain empty as when last k candies are left. JAR is 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

8.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"**

9.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

---

## 10.Problem Statement

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 key = 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.

---

## **11. 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 Wall in units of square feet
4. 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

**Note:** Follow in input and output format as given in above example

## 12.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  $n$ th 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  $d \geq 1000$  metres, then fare=5 INR
- (When calculating fare for others, the calculated fare containing any fraction value should be ceiled. For example, for distance 900m 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

### 13.Problem Statement

There are total n number of Monkeys sitting on the branches of a huge Tree. As travelers offer Bananas and Peanuts, the Monkeys jump down the Tree. If every Monkey can eat k Bananas and j Peanuts. If total m number of Bananas and p number of Peanuts are offered by travelers, calculate how many Monkeys remain on the Tree after some of them jumped down to eat.

At a time one Monkeys gets down and finishes eating and go to the other side of the road. The Monkey who climbed down does not climb up again after eating until the other Monkeys finish eating.

Monkey can either eat k Bananas or j Peanuts. If for last Monkey there are less than k

Bananas left on the ground or less than  $j$  Peanuts left on the ground, only that Monkey can eat Bananas( $<k$ ) along with the Peanuts( $<j$ ).

Write code to take inputs as  $n, m, p, k, j$  and return the number of Monkeys left on the Tree.

Where,  $n$ = Total no of Monkeys

$k$ = Number of eatable Bananas by Single Monkey (Monkey that jumped down last may get less than  $k$  Bananas)

$j$  = Number of eatable Peanuts by single Monkey(Monkey that jumped down last may get less than  $j$  Peanuts)

$m$  = Total number of Bananas

$p$  = Total number of Peanuts

Remember that the Monkeys always eat Bananas and Peanuts, so there is no possibility of  $k$  and  $j$  having a value zero

### Example 1:

Input Values

20

2

3

12

12

### Output Values

Number of Monkeys left on the tree:10

**Note:** Kindly follow the order of inputs as  $n,k,j,m,p$  as given in the above example. And output must include the same format as in above example(Number of Monkeys left on the Tree:)

For any wrong input display INVALID INPUT

---

## 14.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

## 15.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!

---

### 16.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 2100 INR

---

### 17.Explanation:

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**

## Prime Numbers with a Twist

18.Ques. 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.

## Number Series with a Twist – 1

19.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 – – – – – }

## Number Series with a Twist 2

20. 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 is 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<sup>th</sup> 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.

---