

TCS Coding Questions

1. Factorial without Multiplication & Division

You are given a positive integer N . Your task is to compute the factorial of N without using any multiplication $(*)(*)$ or division $(/)(/)$ operators.

Factorial of a number N is defined as: $N! = N \times (N-1) \times (N-2) \times \dots \times 1$ $N! = N \times (N-1) \times (N-2) \times \dots \times 1$.

Input Format

- Get a single integer N , where N is the number for which you need to calculate the factorial.

Output Format

For each test case, output the factorial of the given number N .

2. Identify Row with Most 1

You are given a 2D matrix of size $n \times m$ consisting only of 0's and 1's. Your task is to determine the index of the row that contains the maximum number of 1's. In the case of multiple rows having the same maximum count of 1's, return the index of the first such row.

Input Format

- The first line contains two integers n and m , representing the number of rows and columns in the matrix, respectively.
- The next n lines each contain m integers (00 or 11), representing the elements of the matrix.

Output Format

- Output a single integer, the index (00-based) of the row that has the maximum number of 1's. If all rows contain only 0's, output -1 .

Sample

Input

```
3 4
0 1 0 0
1 1 0 0
0 0 0 1
```

Output

```
1
```

Input

4 4

1 1 1 0

0 1 1 0

0 1 1 1

1 1 1 1

Output

3

3. Count Character Occurrences

You are given two strings, str1 and str2. Your mission is to calculate the total number of occurrences of each unique character of str2 within the string str1. The task is to find the **sum** of occurrences of all unique characters from str2 in str1 and return this total count.

Input Format

- The first line contains the string str1.
- The second line contains the string str2.

Output Format

For each test case, output the total sum of occurrences of characters in str2 found in str1 on a new line.

Sample**Input**

helloworld

do

Output

3

Input

abacabadabacaba

abcd

Output

15

4. Good Number

You are given a number N , and your task is to determine whether it is a "Good Number" or not. A Good Number is defined as a number that is divisible by the sum of its own digits. If the number is divisible by the sum of its digits, it is classified as Good, otherwise, it is classified as Bad.

Input Format

- Each test case contains a single integer N , the number you need to check.

Output Format

For each test case, print "Good Number" if the number is a Good, otherwise print "Bad Number".

Sample

Input

18

Output

Good number

Input

19

Output

Bad number

5. Vehicle Manufacturing

You are tasked with determining the number of two-wheelers and four-wheelers that need to be manufactured based on the given total number of vehicles and the total number of wheels.

You are provided with two integers:

- v : the total number of vehicles (both two-wheelers and four-wheelers).
- w : the total number of wheels for all the vehicles combined.

Your task is to calculate and print how many **two-wheelers** and **four-wheelers** must be manufactured based on the input data. If it's not possible to manufacture such a combination, print $-1\ -1$.

Input Format

- The first line of input will contain a single integer TT , denoting the number of test cases.
- Each test case consists of two lines of input.
 - The first line contains an integer vv — the total number of vehicles.
 - The second line contains an integer ww — the total number of wheels.

Output Format

For each test case,

- If a valid combination of two-wheelers and four-wheelers exists, print two integers:
 - The number of two-wheelers, the number of four-wheelers.
- If no valid combination is possible, print -1.

6. 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=8 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 0 should be pushed towards the end of the array

Input :

8 – Value of N

[4,5,0,1,9,0,5,0] – Element of arr[0] 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

7. Given an **array** of integers of size **N**, the task is to find the first non-repeating element in this array.

Examples:

Input: {-1, 2, -1, 3, 0}

Output: 2

Explanation: The first number that does not repeat is : 2

Input: {9, 4, 9, 6, 7, 4}

Output: 6

8. **Rotate an Array by d – Counterclockwise or Left**

Given an array of integers `arr[]` of size `n`, the task is to rotate the array elements to the left by `d` positions.

Examples:

Input: `arr[] = {1, 2, 3, 4, 5, 6}`, `d = 2`

Output: `{3, 4, 5, 6, 1, 2}`

Explanation: After first left rotation, `arr[]` becomes `{2, 3, 4, 5, 6, 1}` and after the second rotation, `arr[]` becomes `{3, 4, 5, 6, 1, 2}`

Input: `arr[] = {1, 2, 3}`, `d = 4`

Output: `{2, 3, 1}`

Explanation: The array is rotated as follows:

- After first left rotation, `arr[] = {2, 3, 1}`
- After second left rotation, `arr[] = {3, 1, 2}`
- After third left rotation, `arr[] = {1, 2, 3}`
- After fourth left rotation, `arr[] = {2, 3, 1}`

9. Equilibrium Index

Given an array `arr[]` of size `n`, the task is to return an equilibrium index (if any) or -1 if no equilibrium index exists. The equilibrium index of an array is an index such that the sum of all elements at lower indexes equals the sum of all elements at higher indexes.

Note: When the index is at the start of the array, the left sum is 0, and when it's at the end, the right sum is 0.

Examples:

Input: `arr[] = [1, 2, 0, 3]`

Output: 2

Explanation: The sum of left of index 2 is $1 + 2 = 3$ and sum on right of index 2 is 3.

Input: `arr[] = [1, 1, 1, 1]`

Output: -1

Explanation: There is no equilibrium index in the array.

Input: `arr[] = [-7, 1, 5, 2, -4, 3, 0]`

Output: 3

Explanation: The sum of left of index 3 is $-7 + 1 + 5 = -1$ and sum on right of index 3 is $-4 + 3 + 0 = -1$.

10. Print array after it is right rotated K times

Given an Array of size `N` and a value `K`, around which we need to right rotate the array. How do you quickly print the right rotated array?

Examples :

Input: `Array[] = {1, 3, 5, 7, 9}`, `K = 2`.

Output: 7 9 1 3 5

Explanation:

After 1st rotation – `{9, 1, 3, 5, 7}` After 2nd rotation – `{7, 9, 1, 3, 5}`

Input: Array[] = {1, 2, 3, 4, 5}, K = 4.

Output: 2 3 4 5 1

11. Check if an array is subset of another array

Given two arrays a[] and b[] of size m and n respectively, the task is to determine whether b[] is a subset of a[]. Both arrays are not sorted, and elements are distinct.

Examples:

Input: a[] = [11, 1, 13, 21, 3, 7], b[] = [11, 3, 7, 1]

Output: true

Input: a[] = [1, 2, 3, 4, 5, 6], b = [1, 2, 4]

Output: true

Input: a[] = [10, 5, 2, 23, 19], b = [19, 5, 3]

Output: false

12. Given an array of pairs, find all symmetric pairs in it

Two pairs (a, b) and (c, d) are said to be symmetric if c is equal to b and a is equal to d. For example, (10, 20) and (20, 10) are symmetric. Given an array of pairs find all symmetric pairs in it.

It may be assumed that the first elements of all pairs are distinct.

Example:

Input: arr[] = {{11, 20}, {30, 40}, {5, 10}, {40, 30}, {10, 5}}

Output: Following pairs have symmetric pairs

(30, 40)

(5, 10)