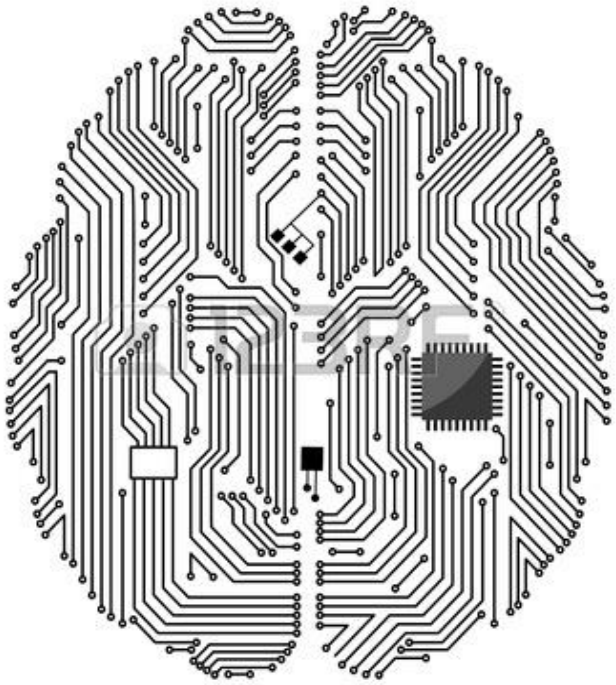
Chittagong University of Engineering & Technology

Onsite programming contest for 13 batch (individual)



Organized by: CUET Computer club

A	**Beautiful String** Input: Standard input Output: Standard output

Samrat is a little boy. He is now learning English alphabet. His mom told him to practice first **N** English alphabet. So, he wrote down English alphabet in a piece of paper in quite random order. After sometime his mom came back. His mom became very angry as she found the string has no beautifulness. Samrat asked his mother, what is beautiful string? His mother said that a string is beautiful if all substring of the string having length less or equal **N** has distinct alphabet. So, Samrat is now trying to make the string beautiful.

As he is lazy, he wanted to know minimum how many alphabet needed to replace to make the string beautiful.

Input:

First line contains an integer T (<=100) which denotes test case number.

Each test case is followed by two line .First line contains two integer L, N where L (0<L<=10^5) is the length of the string and N (1<=N<=8) is the number of alphabet.

Second line contains the string where each character will be one of the first N alphabet and all character are in UPPER-CASE letter. Sum of length of sequence all around the test case will be less than 2x10^6.

Output:

For each test case output a line of one integer **P** where **P** is the minimum number of replace needed to make string beautiful.

Sample Input	Sample Output
2 5 4 ADBBA 6 8 AGGHHC	1 2

Explanation:

In first test case, substring BB (length<=4), has repetition. One possible solution is: ADBCA

B	**Factorial Unreleased** Input: Standard input Output: Standard output

You are given two integers **N** and **A** and a sequence, **P** of prime numbers say, (**P_1**, **P_2**, **P_3**, ..., **P_n**).

You are also given a function, **F =** $\prod_{i=1}^{N}(\frac{A!}{P_i^{f(A,\ P_i)}})^i$ **.**

Where, **f(A, P)** is defined as follows.

```
int f(int A, int P)
{
   int ret = 0;
   int fact_A = 1;
   for(int i = 1; i <= A; i ++)
      fact_A *= i;
   while (fact_A % P == 0)
   {
      fact_A /= P;
      ret ++;
   }
   return ret;
}
```

You have to find the value of the function **F** modulo **10^9+7.**

Input:
The first line of the input file contains an integer **T** (number of test case). In each test case first line is given by **N** and **A** separated by a space. The next line is given the prime sequence **P** (**P_1**, **P_2**, **P_3**, ..., **P_n**).

Output:
For each test case print one line with the value of the function **F** modulo **10^9+7**.

Constraints

$1 \leq T \leq 15$

$1 \leq N \leq 10^5$

$2 \leq A \leq 10^5$

$2 \leq P_i \leq A$

You can assume that the sum of N in whole test file will no more than **10^6.**

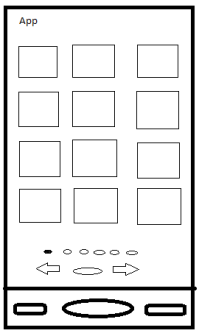
Sample Input	Sample Output
2 3 5 2 2 3 5 11 2 3 5 7 11	216000000 601927535

Note : $\prod_{i=1}^{N}(\frac{A!}{P_i^{f(A,\ P_i)}})^i = (\frac{A!}{P_1^{f(A,\ P_1)}})^1 * (\frac{A!}{P_2^{f(A,\ P_2)}})^2 * (\frac{A!}{P_3^{f(A,\ P_3)}})^3 * \ldots\ldots * (\frac{A!}{P_N^{f(A,\ P_N)}})^N$**.**

C	**Lollipop** Input: Standard input Output: Standard output

Do you know what is Lollipop? In programming contest the easiest problem is known as Lollipop. This problem is about Lollipop. Of course Android Lollipop. Android version 5.0.

Android phones are mostly popular for their easy user interface. They have wonderful menu style. In Android phones applications are occupied in some panels where every panel hold some apps in **r*c** grid. A panel can hold at most **k** apps. You can go to previous or next panel by sliding to left or to right. The panels are circular. That's means if you are in last panel and slide to left then you will go to the first panel and if you are in first panel and slide to right then you will go to the last panel.



Now you are given a phone that runs on Lollipop OS with **n** application numbered from **1** to **n**. This phones can hold at most **k** app in a menu panel. At first you are at home screen. You need one tap to go first menu panel. Then each slide to left or to right takes you next or previous panel. One slide is assumed to be one tap. To open an app you need one tap. You are asked to find minimum number of tap needed to open a desired app. All apps are aligned such that they fill the first panel first and then second panel and so on.

Input:

Input contained one line with three positive number **n**, **k**, **q** (number of apps, number of apps that one panel can hold and the index of the desired app respectively).**1<=n, k, q<=10^9. and q<=n.**

You have to take input till end of file.

Output:

Output contained one line the number of required tap to open the desired app.

Sample Input	**Sample Output**
2 2 2 9 3 5 13 4 9 6 10 6	2 3 4 2

D	**Longest Prime Sequences** Input: Standard input Output: Standard output

In this problem we define**, prime divisor sequence** of prime **P** is a sequence $a_x, a_{x+1}, a_{x+2}\ldots\ldots a_{x+l}$ where each of the element of the sequence is divisible by **P**. Let,

a[]= {1, 2, 6, 3, 9}

Here, {6, 3, 9} is prime divisor sequence of prime 3 of length 3.

Now you are given a number sequence (a_1, a_2, $a_3\ldots\ldots a_n$) and you have to find longest prime divisor sequence.

Input:

First line contains an integer T (<=40) which denotes test case number.

Each test case is followed by two line .First line contains the length of the number sequence($a_1, a_2, a_3\ldots\ldots a_n$) ,n (0<n<=10^5) and second line contains the element a_i (0< a_i <=10^5) where i=1,2,…n. Sum of length of sequence all around the test case will be less than 2x10^6.

Output:

For each test case output a line of two integer **P,L** separated by a space where **P** is prime of the longest prime divisor sequence of length **L**. If there are multiple solution then print the solution which have smallest **P** .If there is no solution then print -1.

Sample Input	**Sample Output**
2 6 2 6 88 3 5 23 5 27 3 5 7 7	2 3 3 2

E	**Tamim Got Worried** Input: Standard input Output: Standard output

It seems that after ICC world cup – 2015 our opener batsman Tamim Iqbal has returned in form. As a proof we can notice it on his performance at recent Bangladesh-Pakistan cricket series - 2015. BCC (Bangladesh Cricket Council) noticed that he practices about 16 to 18 hrs in a day(actually I don't know). But it is so dangerous for his health. For the welfare of both himself and team, BCC has decided to give him a rest. So, BCC is planning how to give Tamim a rest. They got an idea. Everyday BCC give him a problem and tell him, "You can't start practice until you solve this problem". In today's problem he is given two numbers **a** and **b**. In each step he has to take the difference **|a–b|**. Now his new **a** = **min(a,b)** and new **b** = **|a–b|**. In the next step again he takes the difference of new a and b (**| a – b |**). It will go on until the difference is **0**. For example, **a** = **8**, **b** = **5**. If we note all step, it looks like followings.

8-5=3
5-3=2
3-2=1
2-1=1
1-1=0

Here we need five steps to complete the task. In this problem he has to find the total steps until the difference is **0.**

Because of weakness in solving problems, **Tamim got worried**. If he can't solve it then he can't play today. So he wants your help. Since we need him, can you help him?

Input
The first line contains one integer **T** (the number of test cases, $\mathbf{1 \leq T \leq 1000}$). In each case there are two integers **a** and **b** ($\mathbf{1 \leq a, b \leq 10^9}$) in one line separated by a space.

Output
Print the output in the format "Case X: Y" per test case, where X is the number of test case starting from 1 and Y is the total number of steps. There is a blank line between two consecutive outputs.

Sample Input	**Sample Output**
2 8 5 48 24	Case 1: 5 Case 2: 2

F	The Red Planet Input: Standard input Output: Standard output

The Mars is the 4th closest planet of the Sun. It is 225,300,000 km away from the Earth. It is called red planet because its sky is red. NASA is planning to send human on this planet. They send **Curiosity** rover to explore the environment of the planet. The main obstacle to send human on Mars is its distance. Our current fastest rocket needs about 253 days (according to journey time of Curiosity) to reach on Mars. This long journey through space is problematic. To overcome this problem NASA recently run an experiment on new kind of device called **EmDrive** (also **RF resonant cavity thruster**). The device (engine) requires an electrical power source to gain velocity but it does not need any fuel like our current rocket. This experiment gives promising result. Using this device it is possible to reach on Mars only in 70 days. So NASA is preparing for sending human on Mars. Before that NASA undertakes an experiment to know the effect of Mars environment on living body. So they send some bacteria to Mars and observed their behavior. They found a strange behavior on their growth. They notice the flowing behaviors,

1. If the age of the initial bacteria is less than two days then they produce a random amount of new bacteria at the end of the day.
2. If the age of the initial bacteria is at least two days then at the end of the day total bacteria of the current day will produce exact same amount of bacteria as they were in the previous day.
3. There is no effect of age on the bacteria that born on Mars.

The new bacteria that produced by the bacteria of a day will not be counted in total amount of bacteria of that day. It is not necessary that every bacteria will participate in production.

Now as you are a computer expert of NASA, they asks you to run a simulation of the following behaviors and tell them the amount of bacteria on a certain day. They only provides you the amount of bacteria on first day of experiment and number of newborn bacteria on the first day.

Input:

Input contain one line consist of three integer **b** (the number of bacteria on first day), **m** (the number of newborn bacteria on first day), **n** (the day NASA want to know the total amount of bacteria).

1<=b, b+m<=100, 1<=n<=60

You have to take input till end of file.

Output:

Output will contain one line consist of one integer the total amount of bacteria on nth day.

Sample Input	Sample Output
1 1 2 1 3 5 2 3 20	2 14 26073