ACM ICPC
Dhaka Regional Online Preliminary Contest
2018

5th October 2018

You get 13 Pages

10 Problems

A	Welcome **Input:** Standard Input **Output:** Standard Output	

ACM ICPC Dhaka Regional is the biggest programming competition in Bangladesh. Also the most anticipated one as well. Students from all the different universities storm their brains all year in preparation for this competition. You are now taking part in this competition, so a big congratulations to you.

Dhaka site is one of the biggest sites in ACM ICPC. This year almost 1500 teams are participating in the competition. So in celebration to that, we are going to give you an easy problem to start with.

You have to write a program, which will print the line "Welcome to ACM ICPC Dhaka Regional Online Preliminary Contest, 2018" (without quotes).

Note: you can't output anything other than the required output, and each line must end with a newline ('\n'). Take special care about spelling and case. If you alter any of those, you may not get accepted.

For your convenience, we are providing one sample program in C/C++ which prints "Bangladesh". You just have to change the code to your requirement.

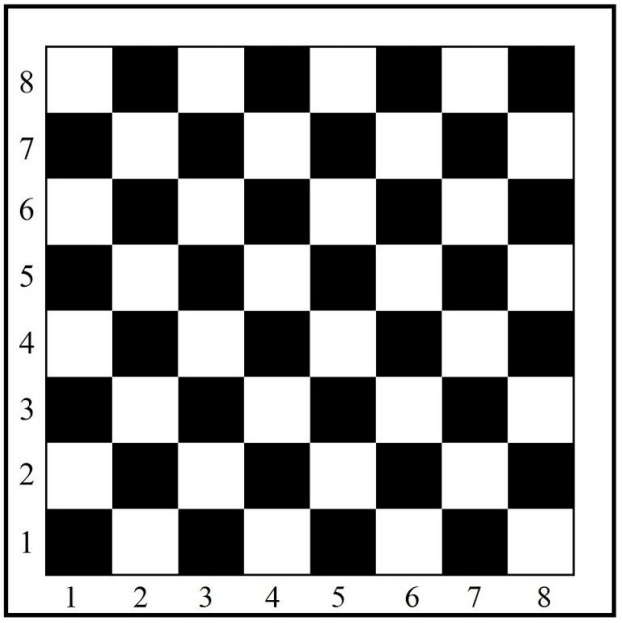
```
#include <stdio.h>
int main()
{
    printf("Bangladesh\n");
    return 0;
}
```

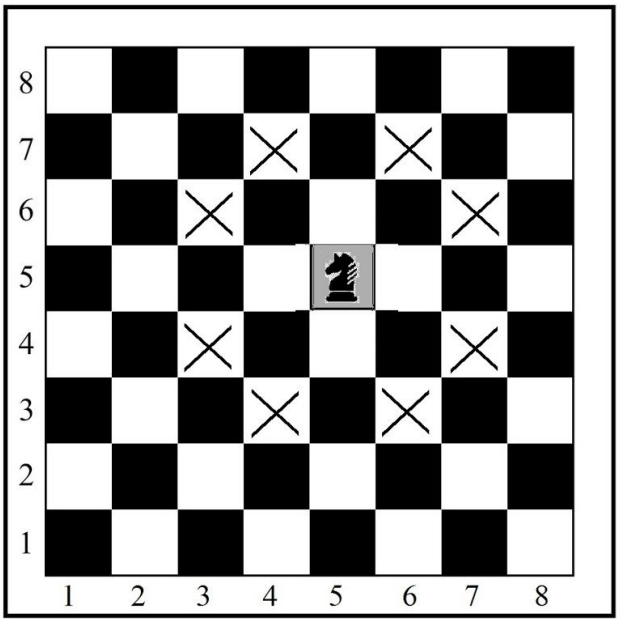
B Boring Chess

Input: Standard Input
Output: Standard Output

"Chess holds its master in its own bonds, shackling the mind and brain so that the inner freedom of the very strongest must suffer." – **Albert Einstein**

Chess is played on a chessboard, a square board divided into 64 squares (eight-by-eight) of alternating color. Knight is a special type of chess piece which moves in an interesting way. Its moves are like "L" shape.
A knight can move from cell (r_1, c_1) to cell (r_2, c_2) if and only if $(r_1 - r_2)^2 + (c_1 - c_2)^2 = 5$. Also, note that a chess piece cannot go out of the board.





In this problem, you are given the current position of a knight **(r, c)**. You have to find, the number of different cells where the knight can go in a single move from its current position.

Input

Input starts with an integer **T (1 ≤ T ≤ 64)** denoting the number of test cases. Following **T** lines will contain two integers **r, c** where **1 ≤ r, c ≤ 8**.

Output

For each test case, output the number of different cells the knight can go in a single move. For more clarity, see sample input/output.

Sample Input	Output for Sample Input
4 5 5 1 2 4 5 1 4	Case 1: 8 Case 2: 3 Case 3: 8 Case 4: 4

C	Odd is real **Input:** Standard Input **Output:** Standard Output	

Given a range **[L, R],** find the number of integers in that range which have an **odd number of odd divisors**. For example, 1 has only one divisor and it is odd, 9 has three divisors {1, 3, 9} and all of them are odd. Meanwhile, 18 has six divisors {1, 2, 3, 6, 9, 18} but three of them are odd. So 1, 9 and 18 have an odd number of odd divisors.

Input

Input will start with a positive integer **T ($T \leq 10^5$)** denoting the number of test cases. Each test case will have two positive integers **L, R ($1 \leq L \leq R \leq 10^{18}$)**, the range.

Output

For each test case, the first line will be the case number in the format **"Case t: x"** without the quotes. Here **t** is the case number starting from 1 and **x** is the number of integers that fall in the range **[L, R]** and have an odd number of odd divisors.

Sample Input	Output for Sample Input
3 1 3 5 10 10 15	Case 1: 2 Case 2: 2 Case 3: 0

D	Prime Friendly Numbers **Input:** Standard Input **Output:** Standard Output	

Given **N,** find the largest number **X,** not greater than **N,** such that **X** is **prime friendly**. A number is called **prime friendly** when it satisfies both of the following conditions:

1. The number itself is a prime.
2. All its digits in **base 10** are also primes. In other words, the number consists of only the digits **2, 3, 5, 7.**

Input

The first line contains an integer **T,** denoting the number of test cases. Each test case contains a single positive integer **N**.

Output

For each test case, output the case number followed by the largest number **X**, not greater **N**. Please refer to the sample input/output section for more clarity of the format.

Constraints:

$1 \leq T \leq 1000$

$2 \leq N \leq 10^{18}$

Sample Input	Output for Sample Input
5 10 100 1000 10000 100000	Case 1: 7 Case 2: 73 Case 3: 773 Case 4: 7757 Case 5: 77773

E	**The End** **Input:** Standard Input **Output:** Standard Output	

You are given a 2-dimensional grid consisting of **N** rows and **M** columns. Rows are numbered from 1 to **N** and columns are numbered from 1 to **M**. You are on the **(1, 1)** cell. You have to go to the **(N, M)** cell.

In each move, you can go to the next cell of the same row or next cell of the same column. For example, if you are on the **(x, y)** cell, then in the next move you can go to the cell **(x, y+1)** or **(x+1, y)**. You can't move outside the grid.

There are also **K** blocked cells on the grid. You can't go to any blocked cell.

Each time, before starting your move, you have **B** magical power. You can enter at most B blocked cell. For the next move after entering the blocked cell on **(x, y)**, you can go either **(x+1, y)** or **(x, y+1)**.

For Example, you are given 3x3 Grid. You have 2 blocks on **(1,2)** and **(1,3)**. You have 1 magical power.

Here are some ways

1. Using one magical power, you can go to the target by path **(1,1) (1,2), (2,2), (3,2), (3,3)**
2. Using zero magical power, you can go to the target by path **(1,1), (2,1), (3,1), (3,2), (3,3)**

But you cannot go to the target cell by path **(1,1), (1,2), (1,3), (2,3), (3,3)** because you need to enter 2 block cells but you have only one magical power.

You have to calculate how many ways you can go from **(1,1)** cell to the target cell **(N, M)**.

N.B.: Two ways are different if there is at least one different cell used in their path from the cell (1,1) to cell (N, M).

Input

Input starts with test case number **T ($1 \le T \le 10$).** For each test case, the first line contains two integers **N ($1 \le N \le 10^6$)** and **M ($1 \le M \le 10^6$)**, denoting the number of rows and columns of the grid. Next line contains two integers **K ($0 \le K \le 100$)** and **B ($0 \le B \le K$)**. Next, each of the K lines contains 2 integers denotes the row **R ($1 \le R \le N$)** and column **C ($1 \le C \le M$)** of the block. You may safely assume that no two blocks will be in the same position. There is no block on the start cell and target cell.

Output

For each test case, output the answer in a single line according to the problem. Since the answer can be very large, simply output the answer modulo **1000000007 ($10^9 + 7$)**.

Sample Input	Output for Sample Input
3 3 3 3 1 1 3 2 2 3 1 3 3 3 0 3 1 2 2 1 3 3 3 2 1 1 2 1 3	6 0 5

F	**Find the Substrings** **Input:** Standard Input **Output:** Standard Output	

You are given a string **S** and **Q** queries. On the **i**th query, you will be given two integers $\mathbf{N_i}$ and $\mathbf{M_i}$. Now for each query, you have to find such strings consisting of lowercase letters whose lengths are at least $\mathbf{N_i}$ and at most $\mathbf{M_i}$ and also those strings are not substrings of **S**.

For example:
You are given **S="abac"**. If a query contains $\mathbf{N_i=3}$ and $\mathbf{M_i=4}$ then according to our problem, **"aba"**, **"bac"** and **"abac"** are not valid as they are substrings of **S** while **"abc"** is a valid string as it is not a substring of **S** and also maintaining the constraints for the length. Now you have to find all other strings which are also valid for this string **S**.

To make the problem a bit simple, you don't have to print all the strings. You will just need to print the number of such strings which are valid. To make the output more simple, print the answer modulo **10^9+7**.

Input

The first line of the input file will be a single integer **T (1 ≤ T ≤ 5),** denoting the number of test cases.
Each test case contains two lines. The first line will contain the string **S (1 ≤ |S| ≤ 1000000)** of lowercase letters and the second line will contain an integer **Q (1 ≤ Q ≤ 100000)**. Each of the next **Q** lines will contain two integers $\mathbf{N_i}$ and $\mathbf{M_i}$ **($1 \le N_i \le M_i \le |S|$).**

Output

For each test case, the first line of the output should contain the case number in the format: **"Case X:",** where **X** is the test case number. Each of the next **Q** lines should contain the answer for the specific query modulo **1000000007** (**10^9+7)**.

Sample Input	Output for Sample Input
1 abcab 5 1 2 2 2 3 5 1 1 1 5	Case 1: 696 673 12355922 23 12356618

N.B. Dataset is large. Use faster I/O methods.

G	Subset with GCD K **Input:** Standard Input **Output:** Standard Output	

Given a set of **N** positive distinct integers and a query value **K**, find if there exists a subset of the integers whose greatest common division equals to **K**.

For example, if the given set is **{1, 6, 2, 9, 8}**, then the answer for the following values of **K** are:

Value of K	Existence of Subset	Possible Subset
1	Yes	{1}
2	Yes	{2,6}
3	Yes	{6,9}
4	No	-

Input

The first line contains a single integer **N (1 ≤ N ≤ 100000)**. The next line contains **N** positive distinct integers separated by whitespace. These **N** integers represent the set as mentioned above.

The third line contains a single integer **Q (1 ≤ Q ≤ 1000)**. The next line contains **Q** distinct positive integers, where each integer is a query **K**.

All values given will be less than or equal to 10^9. See sample I/O for more details.

Output

For each query, output in a single line the character **"Y"** if a subset exists whose GCD equals to **K** or **"N"** if not, without the quotes.

See sample I/O for more details.

Sample Input	Output for Sample Input
5 1 6 2 9 8 4 1 2 3 4	Y Y Y N

H Colorful Balls

Input: Standard Input
Output: Standard Output

Alice has **N** balls arranged in a single line. The balls are either red(**R**), blue(**B**), green(**G**) or white(**W**). They can be represented by a string **S**. Every character of the string is either **R, B, G or W**.

In the beginning, also there are no two balls of the same color side by side (except white ball). For example GGWWB is not a valid string because there are two green balls together. But GWWB is a valid string as there are no two balls of same color side by side except white balls.

Alice needs to paint all the white balls in one of the other three colors in a way that there are no two balls of the same color side by side.

How many ways Alice can paint the balls? Print the solution modulo **1000000007** (**$10^9 + 7$**).

Input

The first line contains the number of test cases **T (1≤T ≤1000)**. In each line of the test cases, there will be a string **S** of length **N (1≤N ≤100000)**.

The total number of character in the input file will be less than **$5 * 10^6$**.

Output

For each test case, print the case number and the answer to the problem.

Sample Input	Output for Sample Input
2 WWG GWGWB	Case 1: 4 Case 2: 2

Explanation for case 1: The four valid ways to color the balls are RBG, BRG, GRG, GBG.

N.B. Dataset is large. Use faster I/O methods.

I	Vugol Search **Input:** Standard Input **Output:** Standard Output	

"**Did You Mean**…" is a search engine function that scans for potential spelling or grammatical errors in user queries and recommends alternative keywords, similar to the auto-correction feature found in mobile messaging services. While the feature is designed to assist users in refining their search results, it has been frequently exploited by **Vugol search** users for comedic purposes. In Vugol search engine, people can search exactly one word at a time.

There are N words in Vugol's database. Each word has its **Smartness** value. If someone does search in the search engine with a word then Vugol search engine calculates the score of each N words with respect to the searched word and suggest the word which has the maximum score with respect to searched word. If the searched word is A, the score is calculated with respect to word B as,

score(A, B) = LCP(A, B) + IsAnagram(A, B) * Smartness value of B
Where:

- LCP(A, B) = length of the longest common prefix of two string A and B.
- IsAnagram(A, B) = If A is anagram of B then 1 else 0.

One or more words from Vugol's database can have the same maximum score with respect to searched word.

You are given Q most searched words in the Vugol search and you already know Vugol's database have N words. For each searched word you have to find the score of the word that Vugol suggests.

Note: An **anagram** of a string is another string that contains the same characters, only the order of characters can be different. For example, "abcd" and "dabc" are anagram of each other.

Input

The first line contains an integer **T,** denoting the number of test cases.
For each test case:

- The first line contains an integer **N** denoting the number of words.
- The second line contains an array of **N**-space separated positive integers denoting Smartness value of words.
- Next **N** lines contain a word.
- Next line contains an integer number **Q**, denoting the number of queries.
- Next, **Q** lines contain a query word.

Constraints

- **1 ≤ T ≤ 10**
- **1 ≤ N, Q**, Smartness value **≤ 10^5**
- A word consists of **only lowercase** English alphabets.
- For **each test case**, the sum of all words' length is within **2 * 10^5**, including query words.

Output

For each query, print the maximum score of a word that Vugol suggests.

Sample Input	Output for Sample Input
1 4 5 2 4 10 google oooogp googoo goloeg 5 google lgooeg glooeg poops goopoo	12 10 11 0 3

N.B. Dataset is large. Use faster I/O methods.

J Yet Another Longest Path Problem

Input: Standard Input
Output: Standard Output

You will be given a graph of **N** nodes connected by **N-1** edges where all the nodes are directly or indirectly connected together. Now we want to modify the graph by making the edges directed. You can visit a node from another one following a path formed by these directed edges. While transforming the graph into a directed one, we also want to minimize the length of the longest possible path in the graph.

You have to determine the direction of each edge to achieve this goal.

Input

There will be several test cases, **T (1 ≤ T ≤ 20)**. For each test case, the first line will contain **N (2 ≤ N ≤ 100000)** denoting the number of nodes. Each of the following **N-1** lines will contain two integers **u, v (1 ≤ u, v ≤ N)** denoting an undirected edge between nodes **u** and **v**.

Output

For each test case, print a line in the format "**Case X:**" where **X** is the integer denoting the test case number starting from 1, then print **N-1** lines, each containing two integers **u** and **v** denoting that there is a directed edge from node **u** to **v** in the graph after the transformation as described above. If there are more than one solutions, print **any** of them, and print in **any order** you want.

Sample Input	Output for Sample Input
2 3 1 2 2 3 4 2 1 1 3 4 1	Case 1: 1 2 3 2 Case 2: 1 2 1 3 1 4

This is a special judge problem.
N.B. Dataset is large. Use faster I/O methods.