The 2011 ACM–ICPC Caribbean National Contests

Real Contest Problem Set

This document contains 9 problems from A to I and it is aimed to be used during the 2011 ACM–ICPC Caribbean National Contests.

Problems' Authors:
Ray Williams Robinson Valiente (UCI, Cuba)
Luis Mariano Savigne Semanat (UCI, Cuba)
Miguel Ángel Covarrubias Sánchez (UG, Mexico)
Yaniel Calviño Cruz (UCI, Cuba)
Yonny Mondelo Hernández (UCI, Cuba)

October 8, 2011.

A – Finding Viruses
Time Limit: 1000 MS

Problem Description

The world is in danger. A new virus has been developed and all antivirus software known so far are useless against this new menace. The ACM has started a secret project to create the ultimate antivirus, a so powerful that it will be able to recognize, not only the whole string of any virus, but also, any possible mutations of it. A mutation of a virus inside a program is any subsequence of its representing string which is a substring inside the string of the program. A subsequence of a string A is any subset of A (including the empty set) in which the characters appear in the same order there are in A. A substring of a string S is any subset (including the empty set) of contiguous characters from S. You have been recruited to collaborate there, due to your great coding skills, since they are stuck in this: they need a program that is able to detect, given a virus string and a software string, what is the longest mutation of the virus present in the software.

Input Details

Input will consist of 2 lines containing 2 lowercase English alphabet strings A ($1 <= |A| <= 10^4$), B ($1 <= |B| <= 10^3$). A will be the software string and B will be the virus string.

Output Details

Output a single line with the length of the longest mutation of B present in A.

Sample Input

```
abacdzz
acexfd
```

Sample Output

```
3
```

B – Square Matrix
Time Limit: 1000 MS

Problem Description

A square matrix of side N is filled with the numbers [1, N^2] in row-major order. After that, it is rotated by 90 degrees K times. Can you guess the number at position R, C after such operation? Row-major order means that the array is filled topmost row first; within each row, leftmost column is filled first.

Input Details

Four space-separated integers N, K, R, C (1 <= N <= 10^6, -1000 <= K<= 1000, 1 <= R, C <= N). When K is negative, the matrix is rotated counter-clockwise, else it is rotated clockwise.

Output Details

Output a single line containing the number at position R, C.

Sample Input	Sample Output
3 1 1 1	7

C – Death Squads
Time Limit: 10000 MS

Problem Description

Young men spend unforgettable moments during the military. Today, they were taken to the field for a shooting practice. There are T targets on the field, which are described by their X, Y coordinates. The shooters are divided into two groups: the first group (#0) is placed along an imaginary straight line that is parallel to the X axis and 100 meters to the south of the southernmost target (the one with the smallest Y coordinate). Similarly, the second group (#1) is placed along an imaginary straight line that is parallel to the Y axis and 100 meters to the west of the westernmost target (the one with the smallest X coordinate).
The first group numbers the targets from 1 to T as they appear while sweeping the field in the West-East direction with a vertical line.
The second group numbers the targets from 1 to T as they appear while sweeping the field in the South-North direction with a horizontal line.
Fire commands are then given in the following fashion:
F (Stands for "Fire!") G (Group number) S (Start) E (End) H (Hits)
For example, the command F 0 1 3 5 means that the first group must aim at the targets in the range [1, 3] according to their numbering and hit each target 5 times.
You are to keep track of the fire commands and answer queries from the General, which are of the type C X Y, meaning "How many times has the target placed at X,Y been hit?". You can assume that the (X, Y) coordinates point to a position of an existing target.
There are neither two targets with the same X coordinate, nor with the same Y coordinate.

Input Details

The first line of input contains two space-separated integers T and C (1 <= T <= 10^5, 1 <= C <= 3*10^5), the number of targets on the field, and the number of commands (Fire or Query), respectively.
T lines follow, each containing two space-separated integers X and Y (-10^6 <= X, Y <= 10^6), the position of a target.
C lines complete the input, the commands given in chronological order, as described previously.
Constraints for the Fire commands: 0 <= G <= 1, 1 <= S <= E <= T, 1 <= H <= 10^5

Output Details

Print a line for each Query command, the answer given to the General. As the answer may be very big, output it modulo 10000007.

Sample Input

```
5 5
1 5
2 3
3 4
4 2
5 1
F 0 2 4 2
F 1 1 5 1
C 3 4
F 1 1 1 1
C 5 1
```

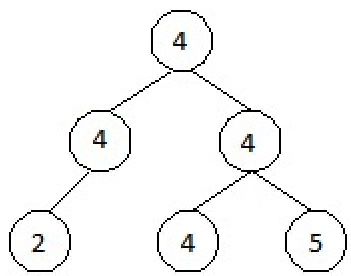
Sample Output

```
3
2
```

D – In-Order Traversal Sequence
Time Limit: 10000 MS

Problem Description

In binary search trees finding the node with a given key is easy. So to make this problem more interesting you have to find all nodes with a given key in in-order. Duplicate keys are allowed. In a tree traversal, from a node the allowed moves are: move to the left subtree (L), move to the right subtree (R) and backtrack to the parent (U).
The in-order traversal first visits the left subtree, then the root and finally visits the right subtree.



The tree created by the sample input.

Input Details

The first line has the key of the root and the number of queries Q. The following Q lines have one of the following queries:
P u L k, insert a left child with key k to node u.
P u R k, insert a right child with key k to node u.
Q k, find all nodes with key k.
Nodes are numbered in the order they appear in the input with the root being number 1. Each P query produces a valid binary search tree. For each Q query, you can assume that there exists at least one node with key k.
Q <= 10^6
number of nodes <= 10^5
0 < k = 10^5
number of nodes with the same key <= 10
depth of tree < 2 log (number of nodes)

Output Details

For each Q query print the smallest sequence of moves which visits the nodes with key **k** from the first to the last. Ordering is given by the in-order traversal. For empty move sequences print a blank line.

Sample Input

```
4 7
P 1 L 4
P 2 L 2
P 1 R 4
P 4 L 4
P 4 R 5
Q 2
Q 4
```

Sample Output

```

URLU
```

E – Bus System
Time Limit: 8000 MS

Problem Description

Recently, a new bus system has been developed by the ACM (Association for Cars and Motorcycles) to minimize the time a passenger needs to move from one place to another. In order to evaluate how effective is the new system, they have gathered the information about the people arriving at the different bus stops. With the collected info, they wish to compute the average time required to complete all journeys, under certain rules. The journey time for a single person is defined as the difference in minutes between the time he comes to the bus stop and the time he gets to his destination.

The new bus system consists of a set of N buses, each one of them with a fixed defined capacity. Since each bus is different, they might not have the same capacity. All buses start the route at the same initial bus stop with a period of T minutes, so the first bus starts at minutes 0, the second one at minute T, the third at minute 2T, and so on. Also, since all buses travel at the same speed, every stop can be described with the time in minutes a bus needs to complete the travel to it from the previous one.

Regarding persons, you may assume the following conditions:

- People will always get the first bus with available capacity.
- Before they can get on a bus, passengers waiting at a bus stop must allow arriving passengers to get off the bus.
- You should not consider the time used for getting up or down to any bus.
- People prefer to travel in groups, so when a group is at some stop, they will get a bus only if they all can get it, otherwise they will wait for a new bus.
- If two or more groups are at the stop when the bus comes, they will get in if possible, according to their arrival times. If two or more groups arrive at the same time, they will break ties according to the group size: smaller groups will get in before larger groups.
- If two or more groups are at the stop when the bus comes, a later group may get in before an earlier one only if there is no capacity for the earlier group.
- You may assume that it is always possible for all people to get to their destinations.
- The ACM is now trying to find a software developer that helps them build the required simulation program. They have contacted you, so this is your chance to show your computer programming skills.

Input Details

Input begins with three integers N (1 <= N <= 100000) and T (1 <= T <= 45), whose meaning has been described above, and M (2 <= M <= 100000) denoting the number of stops. N lines follow with the capacity of the buses in the order they will start the route, one per line. The capacity of every bus is an integer X_i (30 <= X_i <= 100). M - 1 lines follow, where the i-th line indicates the time needed by a bus to complete the travel from the i-th to the (i+1)-th stop. This time will be a value Y_i (1 <= Y_i <= 30). It follows a line with an integer Q (1 <= Q <= 100000), indicating the number of arrivals. Each one of the following Q lines describes an arrival of a group of passengers with 4 integers A, B, C, D where A (1 <= A <= 100) is the number of persons in the group arriving at the stop, B is the arrival time, C (1 <= C <= M) is the stop they're arriving at, and D (C < D <= M) is the destination stop. The arrival time is measured from the time the first bus starts its route. The initial stop has number 1.

Output Details

Output a single number rounded to 3 decimal digits after the decimal point. This number is the average time required to complete the journey for all persons involved. All persons should be considered individually in this computation, regardless of the group they traveled with.

Sample Input

```
2 20 4
50
100
20
20
20
3
1 0 1 2
1 0 1 3
1 0 1 4
```

Sample Output

```
40.000
```

F – Joining the Centers
Time Limit: 20000 MS

Problem Description

In the Math class, all students are excellent. The teacher realizes that they are solving their tasks very easily, so he begins to challenge them with more complex exercises.
Today's problem reads:
You are given N points specified by their X and Y coordinates. Several combinations of such points form triangles. Now, inscribe the resulting triangles inside circles. Let us call S the set of coordinates representing the centers of such circles.
You are to find the minimum-length Hamiltonian path that joins the points in S. You are only allowed to use straight lines between pairs of points in S.
Note: Coordinates may be negative for some calculated centers.

Input Details

The first line of input contains an integer T (1<=T<=10), the number of test cases.
T test cases follow, as described below:
An integer N (3 <= N <= 6), the number of points. N lines follow, each containing two space-separated integers X and Y (0 <= X, Y <= 10^4), the coordinates of a point.

Output Details

Output T lines, each containing the minimum length. Use exactly four decimal places.

Sample Input	Sample Output
1 4 1 2 2 5 3 6 1 3	2.3251

G – Just Another Easy Problem
Time Limit: 1000 MS

Problem Description

Bugi, also known in her classroom as "The Maths Girl", is always testing her classmates' maths skills. Today, she came up with the following question: given an integer N, can you say whether the sum of all natural numbers from 1 to N is a multiple of N? To make things more interesting, she states the number will always be given in hexadecimal notation (i.e. all numbers will be given in a 16-based system and the digits will be numbers from 0 to 9 and uppercase letters from A to F).

Input Details

Input begins with a single integer T (1 <= T <= 1000) denoting the number of test cases. Each test case is described with a single hexadecimal integer N (1 <= N <= 2 ^ 63 - 1) in a line.

Output Details

Output a single line per test case containing **YES** if the required condition holds for this test case. Otherwise the line will say **NO**.

Sample Input	Sample Output
3 1 2 F	YES NO YES

H – Counting Subsequences
Time Limit: 1000 MS

Problem Description

Given a sequence of N numbers, you have to tell how many subsequences of length K exist, such that the greatest common divisor between all its elements is 1.

Input Details

The input consists of several test cases. Each test case consists of two lines:
- the first line has 3 space separated integers N, K and P, with 1 <= K <= N <= 10,000
- the second line has N space separated positive integers, each at most 10,000

Output Details

For each test case, print the number of subsequences modulo the P-th prime (indexed from 0). It is guaranteed that this prime number is less than 10000.

Sample Input

```
4 3 2
2 3 4 5
```

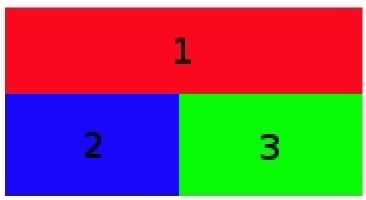
Sample Output

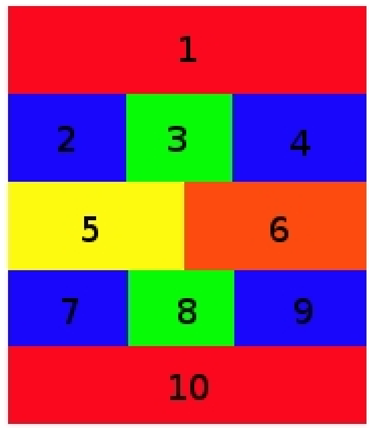
```
4
```

I – Removing Blocks
Time Limit: 3000 MS

Problem Description

Colored Image Blocks (CIB) look like the following images:





A single block may be directly supported by zero, one, or multiple single blocks.
For instance, in the first CIB, the single block 1 is directly supported by single blocks 2 and 3. Single blocks 2 and 3 are not directly supported by any single block.
For a given CIB, you are allowed to remove a single block if and only if there is no other single block directly supported by it.
Can you find the number of ways of removing all the single blocks from a given CIB, according to this rule?

Input Details

Input will be the description of a CIB. The first line of input contains two space-separated integers N and E, where N (1 <= N <= 10) represents the number of single blocks that compose the CIB.
E lines follow, each containing two space-separated integers A and B (1 <= A, B <= N, A != B) meaning that block A is directly supported by block B.

Output Details

For each test case, print the number of subsequences modulo the P-th prime (indexed from 0). It is guaranteed that this prime number is less than 10000.

Sample Input

```
3 2
1 2
1 3
```

Sample Output

```
2
```