The 11th Hunan Collegiate Programming Contest

Online Version

(with more input data and tighter time limits)

30th August, 2015
You get 13 Pages
11 Problems
&
300 Minutes

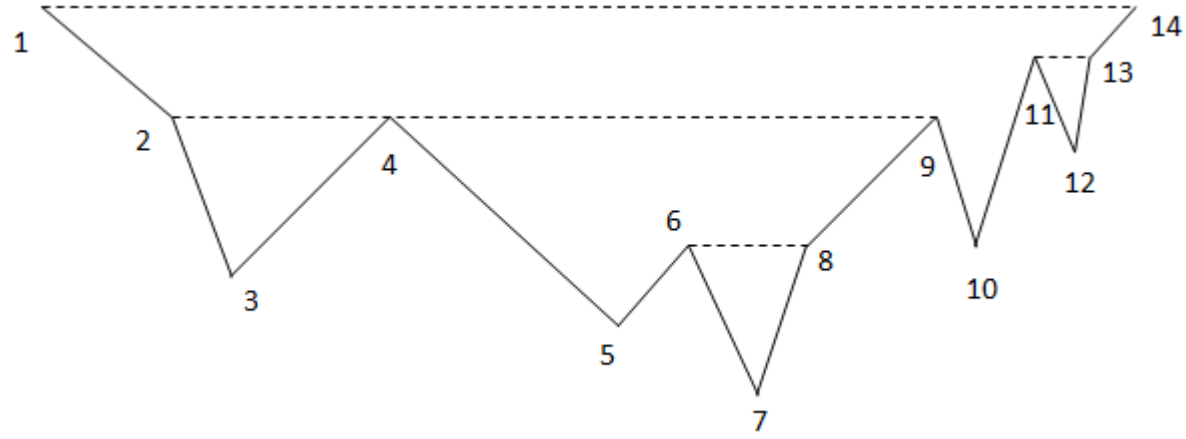
A Aerial Tramway

Input: Standard Input
Output: Standard Output

An aerial tramway, cable car, ropeway or aerial tram is a type of aerial lift which uses one or two stationary ropes for support while a third moving rope provides propulsion. With this form of lift, the grip of an aerial tramway cabin is fixed onto the propulsion rope and cannot be decoupled from it during operations.

-- Wikipedia

You own a park located on a mountain, which can be described as a sequence of *n* points (x_i, y_i) from left to right, where $x_i,y_i>0$, $x_i<x_{i+1}$, $y_i!=y_{i+1}$ (that means there will not be horizontal segments in the mountain skyline), illustrated below (the x-coordinate of p_i is i):



Since the mountain is very sloppy, some aerial tramways across the park would be very helpful. In the figure above, people can go from p_4 to p_9 directly, by taking a tram. Otherwise he must follow a rather zigzag path: p_4-p_5-p_6-p_7-p_8-p_9.

Your job is to design an aerial tramway system. There should be exactly m trams, each following a horizontal segment in the air, between two points p_i and p_j. "Horizontal" means $y_i=y_j$, "in the air" means all the points in between are strictly below, i.e. $y_k<y_i$ for every $i<k<j$. For example, no tram can travel between p_2 and p_9, because p_4 is not strictly below p_2-p_9. However, you can have two trams, one from p_2 to p_4, and one p_4 to p_9. There is another important restriction: no point can be strictly below k or more tramways, because it'll be dangerous. For example, if k=3, we cannot build these 3 tramways simultaneously: p_1-p_{14}, p_4-p_9, p_6-p_8, because p_7 would be dangerous.

You want to make this system as useful as possible, so you would like to maximize the total length of all tramways. For example, if m=3, k=3, the best design for the figure above is p_1-p_{14}, p_2-p_4 and p_4-p_9, the total length is 20. If m=3, k=2, you have to replace p_1-p_{14} with p_{11}-p_{13}, the total length becomes 9.

Input

There will be at most 200 test cases. Each case begins with three integers n, m and k(1<=n,m<=200, 2<=k<=10), the number of points, the number of trams in your design and the dangerous parameter introduced earlier. The next line contains n pairs of positive integers x_i and y_i.(1<=x_i,y_i<=10^5).

Output

For each test case, print the case number and the maximal sum. If it is impossible to have exactly m tramways, print -1.

Sample Input	Output for Sample Input
14 3 3 1 8 2 6 3 4 4 6 5 3 6 4 7 1 8 4 9 6 10 4 11 6 12 5 13 6 14 8 14 3 2 1 8 2 6 3 4 4 6 5 3 6 4 7 1 8 4 9 6 10 4 11 6 12 5 13 6 14 8	Case 1: 20 Case 2: 9

Problemsetter: Rujia Liu, Special Thanks: Md. Mahbubul Hasan, Md. Shiplu Hawlader, Feng Chen

B Bigger or Smaller

Input: Standard Input
Output: Standard Output

Given two floating-point numbers, determine whether the first one is bigger, smaller or the same as the second one. Each floating point number is formatted as

```
[integer part].[fraction part]
```

For simplicity, both of the integer part and fraction part of the given floating point numbers will be non-empty, and the integer part will not have leading zeros. However, fraction part can have trailing zeros, so 0.0 is the same as 0.000.

Input

There will be at most 20 test cases. Each test case contains a single line with two floating-point numbers formatted as above. Each number is a string with no more than 100 characters.

Output

For each test case, print the case number, and one of "Bigger", "Smaller" and "Same".

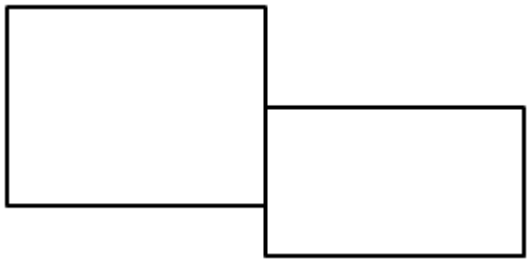
Sample Input	Output for Sample Input
1.0 2.0 0.00001 0.00000 0.0 0.000	Case 1: Smaller Case 2: Bigger Case 3: Same

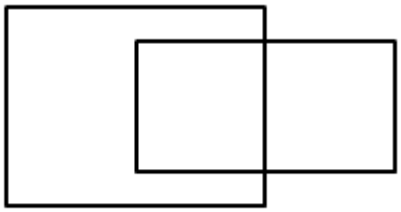
Problemsetter: Rujia Liu, Special Thanks: Md. Mahbubul Hasan, Feng Chen

C Common Area

Input: Standard Input
Output: Standard Output

Given two simple polygons, your task is to determine whether they have a non-empty common area. Note that the two rectangles in figure (a) share a segment, but no common area at all.





By "simple polygon", we mean the polygons will not be self-intersecting or self-touching, and will not have duplicated vertices or adjacent collinear segments.

Note: be sure to test your program with many special cases.

Input

There will be at most 100 test cases. Each test case contains two lines, one for each polygon. Each polygon begins with an integer n (3<=n<=100), the number of vertices, then n pairs of integers (x,y) (-1000<=x,y<=1000), the vertices of the polygon, in counter-clockwise order.

Output

For each test case, print the case number and one of "Yes" or "No".

Sample Input	Output for Sample Input
4 0 0 2 0 2 2 0 2 4 2 0 4 0 4 2 2 2 4 0 0 2 0 2 2 0 2 4 1 0 3 0 3 2 1 2	Case 1: No Case 2: Yes

Problemsetter: Rujia Liu, Special Thanks: Md. Mahbubul Hasan, Md. Shiplu Hawlader, Feng Chen

D Defeat the Wrong Program

Input: Standard Input
Output: Standard Output

Find a cell in an *n*m* grid such that the sum of integers in the "cross shape" (a cell and all the cells in the same row or column) is maximized. **If the answer is not unique, you can print any solution.**

In the example below:

5	5	5	5
1	1	5	1
1	1	5	1
1	1	5	1

The best solution is (1,3), i.e. the intersection of the first row from above, and the 3rd column from left. The sum of integers in the cross-shape is 5*7 = 35.

The problem above is from a local programming contest. After the contest data has been generated, one of the judges has come up with the following algorithm:

> First, find a row r(1<=r<=n) with the maximum sum, then find a column c(1<=c<=m) with the maximum sum, and finally output (r,c). If there is a tie, print the smallest r and/or c.

Obviously, this algorithm does NOT guarantee a correct answer, but to their surprise, this program passed most judge data! Could you identify the "weak" data so that the judges can improve these data?

Input

There will be at most 100 test cases. Each case begins with two integers n, m (1<=n<=500, 1<=m<=500), the number of rows and columns. Then n lines follow, each containing m integers between 1 and 100. The whole input size does not exceed 2MB.

Output

For each test case, print the case number and "Weak" or "Strong".

Sample Input	Output for Sample Input
4 4 5 5 5 5 1 1 5 1 1 1 5 1 1 1 5 1 5 4 2 5 1 1 1 1 9 1 1 1 1 1 1 1 1 1 1 1 1 1	Case 1: Weak Case 2: Strong

Problemsetter: Rujia Liu, Special Thanks: Md. Mahbubul Hasan, Feng Chen

E Easy Graph Problem?

Input: Standard Input
Output: Standard Output

Given a grid maze with n rows and m columns, each cell is either an obstacle or has a cost associated. Your task is to go from (r1,c1) to (r2,c2) with minimum cost. In each step, you can go up, down, left or right, but not diagonally. Of course you cannot go into an obstacle or go out of the maze. The total cost of a path equals the sum of costs in each cell in the path.

In the picture below, all shaded cells are obstacles. If you go along A->B->D->F->E, the cost is 10+3+6+14+8=41. **Note that if you visit a cell twice, the cost is added twice.**

	A	B	G
	C	D	H
	E	F	

7	10	3	9
	45	6	2
	8	14	
21	1		

To make things more interesting, you need to answer an additional question: what if you **must make a turn** (left turn, right turn or U-turn) in each step. For example, if you go from A to B in the first step, you can go from B to D or A, but not G. In the example above, the best route is A->B->D->H->D->F->E, the total cost is 10+3+6+2+6+14+8=49. Note that D is visited twice.

Input

There will be at most 10 test cases. Each test case begins with a line containing six integers n, m, r1, c1, r2, c2 (2<=n,m<=500, 1<=r1,r2<=n, 1<=c1,c2<=m). The next n lines describe the maze. Each cell is either a positive integer between 1 and 100 or an asterisk "*". Note that neither the starting cell nor the ending cell can be an obstacle.

Output

For each test case, print the case number and two integers, the first one is the answer to the "normal case", and the next one is the answer to the "interesting case". If there is no route in either case, the corresponding integer should be -1.

Sample Input

```
4 4 1 2 3 2
7 10 3 9
* 45 6 2
* 8 14 *
21 1 * *
2 4 1 1 1 4
1 2 3 4
9 * * 9
2 4 1 1 1 4
1 * 3 4
9 9 * 9
```

Output for Sample Input

```
Case 1: 41 49
Case 2: 10 -1
Case 3: -1 -1
```

Problemsetter: Rujia Liu, Special Thanks: Md. Mahbubul Hasan, Feng Chen

F Factorial Division

Input: Standard Input
Output: Standard Output

Given two positive integers n, m, find out n!/m!, where n!= 1*2*3*...*n (n>=1).

For example, if n=6, m=3, 6!/3!=720/6=120.

Easy, right? Now let's do the reverse: given k=n!/m!, find out the pair (n,m) (n>m>=1).

If there is more than one solution, n should be as small as possible. For example, if k=120, the answer should be n=5 and m=1, not n=6 and m=3, because 5!/1!=6!/3!=120, and 5<6.

Input

There will be at most 100 test cases. Each test case contains one integer k ($1<=k<=10^9$).

Output

For each test case, print two integers n and m. If there is no solution, print "Impossible". If there is more than one solution, n should be as small as possible.

Sample Input	Output for Sample Input
120 1 210	Case 1: 5 1 Case 2: Impossible Case 3: 7 4

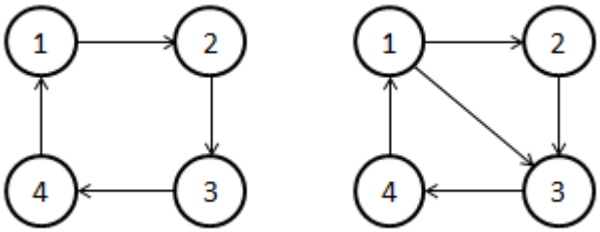
Problemsetter: Rujia Liu, Special Thanks: Md. Mahbubul Hasan, Feng Chen

G Graph Guessing

Input: Standard Input
Output: Standard Output

There is a strongly-connected graph (i.e. you can reach any node from any other node) with n nodes and m edges. I will choose some of the edges to make another strongly connected graph. Your task is to guess that graph. Too difficult, right? Don't worry, you only need to guess k edges. If all the edges exist in my graph, you win. I promise that from all possible graphs, the answer will be chosen uniformly. **The original graph will not have self-loops or duplicated edges**.

You already have a guess, but you are a bit unsure. Why not write a program to calculate the probability you win? For example, if n=4, m=5, the original graph has 5 edges: 1->2, 2->3, 3->4, 4->1, 1->3, there are only two possible answers:



If k=2, the best way is to guess edge 1->2 and 2->3 (or 1->2 and 3->4 etc.) which will guarantee a win. But if you would like to risk by guessing edges 1->3 and 2->3, the probability you win is 0.5.

Input

There will be at most 10 test cases. Each case begins with two integers n, m (3<=n<=15, 2<=m<=50). Each of the following m lines contains two different integers u, v (1<=u,v<=n), that means u->v is in the original graph. Edges are numbered 1 to m in the same order they appear in the input. The last line begins with an integer k (1<=k<=m) and k different integers, the edges you guess.

Output

For each test case, print the case number and the probability you win. Absolute error of 10^{-4} is allowed.

Sample Input	Output for Sample Input
4 5 1 2 2 3 3 4 4 1 1 3 2 1 2 4 5 1 2 2 3 3 4 4 1 1 3 2 5 2	Case 1: 1.0000 Case 2: 0.5000

Problemsetter: Rujia Liu, Special Thanks: Md. Mahbubul Hasan, Shiplu Hawlader
Based on a problem in Chinese IOI Training camp, by Lijie Chen

H

Hehe

Input: Standard Input
Output: Standard Output

There is a Chinese joke: "Chat ends with hehe". Given a chat log, count how many conversations end with a sentence containing the word "hehe" (or variations, see below).

To check whether the sentence contains "hehe", first replace any non-alphabetic characters with spaces, convert the characters to lower-case and extract a list of words. For example, "Hi! Are you OK?" becomes a list of 4 words: hi, are, you, ok. **Note that the list of words could be empty.**

Note that some people prefer to use "hehehe" or "hehehehe" instead of "hehe", so a word formed by n(n>1) copies of "he" should be regarded as a variation of "hehe". However, there are some other words which contain "hehe" as a consecutive substring. Don't consider them!

Input

There is only one test case. Each line is a record formatted as

```
Name1->Name2: sentence.
```

Each line will have at most 1000 characters, and there will be at most 100 lines.

Output

Print the percentage (rounded to the nearest integer) of conversations ended with "hehe" or its variants. The test data will be carefully chosen so that the answer will not be equally near to two integers.

Sample Input	Output for Sample Input
A->B: Hello! A->C: Hi! B->A: Hehe B->D: Hei! D->B: How are you? A->C: Hi??? A->C: Are you there? B->D: Hehehei! D->B: What does hehehei mean? F->E: I want to hehehehehe yah.	50%

Explanation

Converstation between A and B ends with "Hehe".
Converstation between A and C ends with "Are you there?".
Converstation between B and D ends with "What does hehehei mean?".
Converstation between E and F ends with "I want to hehehehehe yah".
Only the first one and the last one ends with "hehe" (and variants), so the answer is 50%.

Problemsetter: Rujia Liu, Special Thanks: Md. Mahbubul Hasan, Feng Chen

I Internet of Lights and Switches

Input: Standard Input
Output: Standard Output

You are a fan of "Internet of Things"(IoT), so you build a nice Internet of Lights and Switches in your huge mansion. Formally, there are *n* lights and *m* switches, each switch controls one or more lights, i.e. pressing that switch flips the status of those lights (on->off, off->on).

Initially, all the lights are on. Your task is to count the number of ways to turn off all the lights by pressing some *consecutive* switches. There is only one restriction: the number of switches you pressed should be between a and b (inclusive).

Input

There will be at most 20 test cases. Each test case begins with a line containing four integers n, m, a, b (2<=n<=50, 1<=a<=b<=m<=300000). Each of the following m lines contains a 01 string of length n. The i-th character is 1 if and only if that switch controls the i-th light. The size of the whole input file does not exceed 8MB.

Output

For each test case, print the case number, and the number of ways to turn off all the lights.

Sample Input	Output for Sample Input
2 4 1 4 01 10 11 00 2 4 3 3 01 10 11 00 6 3 1 3 101001 010110 101001	Case 1: 3 Case 2: 0 Case 3: 2

Problemsetter: Rujia Liu, Special Thanks: Feng Chen, Md. Mahbubul Hasan

J Just Another Easy Problem

Input: Standard Input
Output: Standard Output

Given a four-digit integer n, your task is to count the number of ways to make it a square number by changing exactly one digit (note that you can't change the first digit to 0). For example, if n=7844, there are two ways: $3844=62^2$ and $7744=88^2$.

Input

The first line of integer contains one integer T (1<=T<=1000), the number of test cases. Each test case contains a single integer n (1000<=n<=9999).

Output

For each test case, print the case number and the number of ways to make it a square integer by changing exactly one digit.

Sample Input	Output for Sample Input
2 7844 9121	Case 1: 2 Case 2: 0

Problemsetter: Rujia Liu, Special Thanks: Md. Mahbubul Hasan, Feng Chen

K Keep Fit!

Input: Standard Input
Output: Standard Output

You're designing a game console with a special board which can evaluate how balance people are. After the user stands on the board, it can record the movement of the user's "center of gravity".

Technically, the record is a sequence of n points on the 2D plane (the user's "center of gravity" projected to the game board), where the origin (0,0) is the center of the game board. Samples are taken every 0.01 second, so if the user stands on it for one minute, your database gets 6000 sample points.

In order to know better about his balancing status, the user can ask the game console some questions. Each question (i,j) means: count how many pairs of sample points, chosen from the interval between the i-th sample and the j-th sample (inclusive), whose Manhattan distance is **no more than** d, where d is the preset *balance threshold* parameter in the system.

Your task is to write a program that can answer the questions. Note that you don't have to answer the questions one by one. **You can read all the questions first, and then answer them.**

Input

There are no more than 3 test cases. The first line contains three integers n, d, q($1<=n<=200000$, $1<=d<=10^8$, $1<=q<=1000$), the number of points, the balance threshold and the number of queries. The next n lines contain the coordinates (x,y) ($|x|,|y|<=10^8$) of the sample points, in order. The points are numbered 1~n. The next q lines contain the questions (i,j) ($1<=i<=j<=n$).

Output

For each test case, print the case number in the first line, then the answers of the questions, one on each line.

Sample Input

```
5 1 2
0 0
1 0
3 0
2 1
2 0
2 4
1 5
5 2 2
0 0
1 0
3 0
2 1
2 0
2 4
1 5
```

Output for Sample Input

```
Case 1:
0
4
Case 2:
3
8
```

Problemsetter: Rujia Liu, Special Thanks: Md. Mahbubul Hasan