The CENTRE for EDUCATION
in MATHEMATICS and COMPUTING

2009 Canadian Computing Competition: Senior Division

Sponsor:

University of
Waterloo

Canadian Computing Competition
Student Instructions for the Senior Problems

1. You may only compete in one competition. If you wish to write the Junior paper, see the other problem set.

2. Be sure to indicate on your **Student Information Form** that you are competing in the **Senior** competition.

3. You have three (3) hours to complete this competition.

4. You should assume that

   - all input is from a file named `sX.in`, where `X` is the problem number ($1 \leq$ `X` ≤ 5).
   - all output is to the screen

   Since your program will read from a file, there is no need for prompting. Be sure your output matches the output in terms of order, spacing, etc. IT MUST MATCH EXACTLY!

5. Do your own work. Cheating will be dealt with harshly.

6. Do not use any features that the judge (your teacher) will not be able to use while evaluating your programs.

7. Books and written materials are allowed. Any machine-readable materials (like other programs which you have written) are *not* allowed. However, you are allowed to use "standard" libraries for your programming languages; for example, the STL for C++, java.util.*, java.io.*, etc. for Java, and so on.

8. Applications other than editors, compilers, debuggers or other standard programming tools are **not** allowed. Any use of other applications will lead to disqualification.

9. Please use file names that are unique to each problem: for example, please use `s1.pas` or `s1.c` or `s1.java` (or some other appropriate extension) for Problem S1. This will make the evaluator's task a little easier.

10. Your program will be run against test cases other than the sample ones. Be sure you test your program on other test cases. Inefficient solutions may lose marks for some problems, especially Problems 4 and 5. Be sure your code is as efficient (in terms of time) as possible.

11. Note that the top 2 Senior competitors in each region of the country will get a plaque and $100, and the schools of these competitors will also get a plaque. The regions are:

    - West (BC to Manitoba)
    - Ontario North and East
    - Metro Toronto area

- Ontario Central and West
- Quebec and Atlantic

12. If you finish in the top 20 competitors on this competition, you will be invited to participate in CCC Stage 2, held at the University of Waterloo in May 2009. Should you finish in the top 4 at Stage 2, you will be a member of the Canadian IOI team at IOI 2009, held in Bulgaria. Note that you will need to know C, C++ or Pascal if you are invited to Stage 2. But first, do well on this contest!

13. Check the CCC website at the end of March to see how you did on this contest, how the problems were meant to be solved, and to see who the prize winners are. The CCC website is:

`www.cemc.uwaterloo.ca/ccc`

Problem S1: Cool Numbers

Problem Description
Eric likes interesting numbers like 64. It turns out that 64 is both a square and a cube, since $64 = 8^2$ and $64 = 4^3$. Eric calls these numbers *cool*.

Write a program that helps Eric figure out how many integers in a given range are cool.

Input Description
On the first line of input, you are given an integer a such that $a \geq 1$ and $a \leq 10^8$. On the second line of input, you are given an integer b such that $a \leq b$ and $b \leq 10^8$.

Output Description
The output should be the number of cool numbers in the range a to b (inclusively: that is, a and b would count as cool numbers in the range if they were actually cool).

Sample Input 1
```
1
100
```

Output for Sample Input 1
```
2
```

Sample Input 2
```
100
1000
```

Output for Sample Input 2
```
1
```

Problem S2: The Lights Going On and Off

Problem Description

For your birthday, you have been given a grid of R ($1 < R < 30$) rows of lights, with each row containing L ($1 \leq L < 8$) lights. Lights can be in one of two states: on or off. For this question, the topmost row is row R, and the bottom-most row is row 1. Also, beside all rows except the topmost row (row R), there is a button which can be pushed.

Pushing the button beside row k ($1 \leq k < R$) will peform an "exclusive-or" operation on each light of row k, which is described below. Consider column i in row k, where $1 \leq i \leq L$. If the lights in column i of row k and column i of row $k+1$ are both the same (i.e., both on, or both off), then pushing the button beside row k will cause the light in column i of row k to be off. If the lights in column i of row k and column i of row $k+1$ are different (i.e., one is on, and the other is off), then pushing the button beside row k will cause the light in column i of row k to be on. An example is shown below, for $L = 4$:

Column numbers	1	2	3	4
Row $k+1$	on	on	off	off
Row k before button pushed	on	off	on	off
Row k after button pushed	off	on	on	off

You are told which lights are initially on and which are initially off. You must calculate how many different light patterns are possible for the bottom row by any sequence of button pushes.

Input Description

The first line of input will contain the integer R, the number of rows. The second line of input will contain the integer L, the number of lights per row. The next R lines of input will contain L integers, where the integer will either be 0 (to indicate "off") or 1 (to indicate "on"). Pairs of consecutive integers will be separated by one space character. These R lines will be given in top-down order: that is, the third line of input will be the description of row R, the fourth line will be $R-1$, and so on, until the last line describes the bottom row.

Output Description

Output the number of possible light patterns of the bottom row.

Sample Input

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

Output for Sample Input

```
4
```

Problem S3: Degrees of Separation

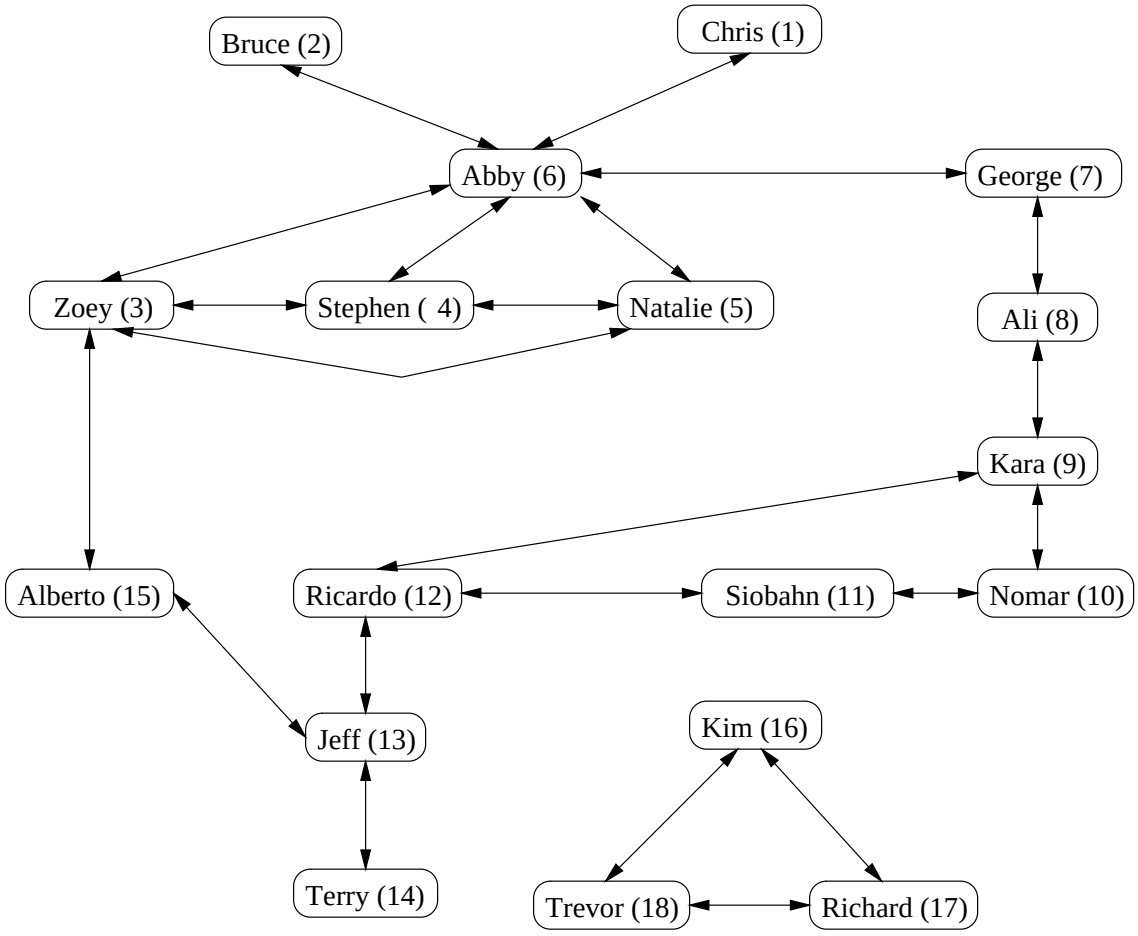
Problem Description

The main socializing tool for students today is Facebook. There are many interesting computational questions connected to Facebook, such as the "degree of separation" between two people.

For example, in the diagram below, there are many different paths between Abby and Alberto. Some of these paths are:

- Abby → Zoey → Alberto
- Abby → Natalie → Zoey → Alberto
- Abby → George → Ali → Kara → Richardo → Jeff → Alberto

The shortest path between Abby and Alberto has two steps (Abby → Zoey, and Zoey → Alberto), so we say the degree of separation is 2. Additionally, Alberto would be a friend of a friend of Abby.



You can assume an initial configuration of who is friends with who as outlined in the diagram above. You will need to store these relationships in your program. These relationships can change though, and your program needs to handle these changes. In particular, friendships can begin, possibly with new people. Friendships can end. You should be able to find friends of friends and determine the degree of separation between two people.

Input/Output Description

Your program will read in six possible commands, with the action to be performed by your program outlined below. You may assume that x and y are integers, with $x \neq y$, $x \geq 1$, $y \geq 1$, $x < 50$ and $y < 50$. You may also assume that instructions (`i`, `d`, `n`, `f`, `s`, `q`) occur one per line and parameters (zero, one or two integers) occur one per line.

- `i` x y – make person x and person y friends. If they are already friends, no change needs to be made. If either x or y is a new person, add them.
- `d` x y – delete the friendship between person x and person y.
- `n` x – output the number of friends that person x has.
- `f` x – output the number of "friends of friends" that person x has. Notice that x and direct friends of x are not counted as "friends of friends."
- `s` x y – output the degree of separation between x and y. If there is no path from x to y, output `Not connected`.
- `q` – quit the program.

Sample Interaction

Input	Output	Explanation

`i`
`20`
`10`
	(no output)	Inserting a friendship causes no output.

`i`
`20`
`9`
	(no output)	Inserting a friendship causes no output.

`n` `20`	`2`	Person 20 has two friends (10 and 9)
`f` `20`	`3`	The friends of friends of 20 are 8, 11, 12.

`s`
`20`
`6`
	`4`	The shortest path is $20 \to 9 \to 8 \to 7 \to 6$.

`q`	(no output)	Program quits.

Problem S4: Shop and Ship

Problem Description

In Doubleclickland, there are N cities ($N \leq 5,000$), with each city having various trade routes to other cities. In total, there are T trade routes ($0 \leq T \leq 25,000,000$). in Doubleclickland. For each trade route between two cities x and y, there is a transportation cost $C(x, y)$ to ship between the cities, where $C(x, y) \geq 0$, $C(x, y) \leq 10,000$ and $C(x, y) = C(y, x)$. Out of the N cities, K ($1 \leq K \leq N$) of these cities have stores with really nice pencils that can be purchased on-line. The price for each pencil in city x is P_x ($0 \leq P_x \leq 10,000$).

Find the minimal price to purchase one pencil on-line and have it shipped to a particular city D ($1 \leq D \leq N$) using the cheapest possible trade-route sequence. Notice that it is possible to purchase the pencil in city D and thus require no shipping charges.

Input Description

The first line of input contains N, the number of cities. You can assume the cities are numbered from 1 to N. The second line of input contains T, the number of trade routes. The next T lines each contain 3 integers, x y $C(x, y)$, to denote the cost of using the trade route between cities x and y is $C(x, y)$. The next line contains the integer K, the number of cities with a store that sells really nice pencils on-line. The next K lines contains two integers, z and P_z, to denote that the cost of a pencil in city z is P_z. The last line contains the integer D, the destination city.

Output Description

Output the minimal total cost of purchasing a pencil on-line and shipping it to city D.

Sample Input

```
3
3
1 2 4
2 3 2
1 3 3
3
1 14
2 8
3 3
1
```

Output for Sample Input

```
6
```

Problem S5: Wireless

Problem Description

Bob is sitting at home with his computer. He would like to experience more social interaction, so he is planning a trip to a coffee shop with his computer.

Bob has lots of data about wireless networks and coffee shops in the city. In Bob's city, there is one coffee shop at *every* intersection of streets. Specifically, Bob happens to live in a city with M streets ($1 \leq M \leq 30,000$) that run east and west, and N streets ($1 \leq N \leq 1,000$) that run north and south. As an added benefit, the distance between consecutive parallel streets is 1 metre (it is a very compact city).

It also turns out that inside K ($1 \leq K \leq 1,000$) of the coffee shops, there is a wireless network station. Each wireless network station will have a particular bitrate B ($1 \leq B \leq 1,000$) and can reach R metres ($1 \leq R \leq 30,000$) from the coffee shop. In other words, a wireless network station from one coffee shop forms a circle with radius R centered at that particular coffee shop. Moreover, if someone is at distance R, the wireless network would be available, but if someone is at a distance larger than R, they cannot access that wireless point.

You can assume that each coffee shop has at most one wireless network stationed in it, but that multiple wireless networks may be available while sitting in that one coffee shop, due to the proximity of other wireless network stations.

Bob has a special device in his computer that can use all of the available bitrates of as many wireless networks as he can connect to.

Bob would like to find out the maximum bitrate he can obtain, and how many coffee shops would have that maximum capacity.

Input Description

On the first line of input, you will be given the integer M, the number of east-west streets. On the second line of input, you will be given the integer N, the number of north-south streets. On the third line of input, you will be given the integer K, the number of coffee shops with a wireless network. On the next K lines, you will have 4 integers per line. The first integer x on each line represents the north-south street on which the coffee shop is located, where $1 \leq x \leq N$. The second integer, y, on each line represents the east-west street on which the coffee shop is located, where $1 \leq y \leq M$. The third integer, R, on each line represents the radius of the wireless network from this particular coffee shop. The fourth integer, B, on each line represents the bitrate of the wireless network from this particular coffee shop.

Output Description

The output will be two lines long. The first line of output will be the integer representing the maximum bitrate that can be obtained over all coffee shops. The second line of output will be the number of coffee shops where this maximum bitrate can be obtained.

Sample Input

```
3
5
3
1 3 2 5
3 1 2 7
5 1 1 5
```

Output for Sample Input

```
12
5
```

Explanation of Sample Input and Output

In the figure below, notice that the five coffee shops/intersections marked with a dark circle have total bitrates of 12.