

2016 Canadian Computing Olympiad

Day 1, Problem 1

Field Trip

Time Limit: 2 seconds

Problem Description

As a special treat for your kindergarten class, you're taking them on a field trip to a magical place of wonder.

Your class has N students, numbered from 1 to N for convenience. There are M direct, two-way friendships that exist between the students. Each student is friends with at most two other students.

Aside from the M direct friendships, students may also be acquainted with one another. Two students i and j are acquaintances if they're friends, or if there exists a third student k who is an acquaintance of both students i and j . For example, if $(1, 2)$, $(2, 3)$, $(3, 4)$ and $(4, 5)$ are pairs of students with a direct friendship, then person 1 and person 5 are acquaintances.

You're getting ready to order buses for the trip, but there are two issues. Firstly, the transportation company insists that every bus you order must be filled exactly to its capacity of K students. They won't allow you to order a bus if you intend to put fewer than K students on it! Secondly, the students are picky about their travelling conditions. Each student i will refuse to get on a bus unless both of the following conditions are met:

1. All of the other students getting on that bus are acquaintances of student i ;
2. All of student i 's acquaintances are getting on that bus.

Unfortunately, it looks like you might not be able to bring your whole class on this trip after all. However, you'll do whatever it takes to get as many students as possible on buses. As it turns out, "whatever it takes" may involve putting an end to a friendship or two, for the greater good. You may choose to sever 0 or more of the M friendships amongst the students, which will of course also have an effect on which students are acquainted with one another.

Determine the maximum number of students which can be brought on the trip, such that they're loaded onto buses with exactly K students each, and every student is satisfied with their bus allocation. Furthermore, since you're feeling generous, determine the minimum number of friendships which you can sever in order to be able to bring that many students along.

Input Specification

The first line contains three space-separated integers N , M and K ($1 \leq N \leq 10^6$; $0 \leq M \leq 10^6$; $1 \leq K \leq N$).

The next M lines contain information about the friendships. That is, each of these M lines contain two space-separated integers A_i and B_i ($1 \leq i \leq M$) describing that students A_i and B_i are friends ($1 \leq A_i, B_i \leq N$, $A_i \neq B_i$). Note that no friendship is specified twice (that is, no two unordered friendship pairs are equal to one another).

For 3 of the 25 marks available, $N \leq 1000$.

Output Specification

The output consists of two space-separated integers printed on one line. The first integer is the the maximum number of students which can be brought on the trip. The second integer is the minimum number of friendships which must be severed in order to bring that many students.

Sample Input

```
8 5 2
1 4
8 2
4 5
6 2
3 5
```

Output for Sample Input

```
6 2
```

Explanation for Output for Sample Input

If the friendships between student pairs (8,2) and (4,5) are severed, then 3 buses can be filled as follows:

- Bus 1: Students 1 and 4
- Bus 2: Students 2 and 6
- Bus 3: Students 3 and 5

2016 Canadian Computing Olympiad

Day 1, Problem 2

Splitting Hares

Time Limit: 15 seconds

Problem Description

As you know, some bunnies are good bunnies, and some bunnies are bad bunnies.

You are given the location of all the bunnies, and their “goodness” weight (a positive integer for good bunnies and a negative integer for bad bunnies). No two bunnies are at the same location. Divide them into two groups using a straight line such that the sum of the “goodness” of the bunnies on one side of the line is as large as possible. A bunny on the line is counted in the sum of the weights on both sides of the line.

Input Specification

The first line contains N ($2 \leq N \leq 4000$), the number of bunnies. The next N lines contain three space-separated integers: $x_i y_i w_i$, which indicates that at the point (x_i, y_i) there is a bunny with a goodness weight of w_i ($-1\,000\,000 \leq x_i \leq 1\,000\,000$; $-1\,000\,000 \leq y_i \leq 1\,000\,000$; $-10\,000 \leq w_i \leq 10\,000$). The locations (x_i, y_i) will be distinct (i.e., there is no other $j \neq i$ such that $(x_i, y_i) = (x_j, y_j)$).

For 5 of the 25 marks available, $N \leq 200$ and no three locations are collinear.

For an additional 10 of the 25 marks available, no three locations are collinear.

Output Specification

Output the maximum sum of weights that is possible by drawing a straight line and picking all the bunnies which are on one side of that line.

Sample Input

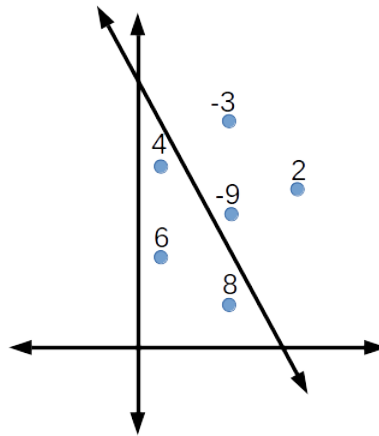
```
6
1 8 4
1 4 6
7 7 2
4 10 -3
4 6 -9
4 2 8
```

Output for Sample Input

```
18
```

Explanation for Output for Sample Input

Take the bunnies with goodness weights of 4, 6 and 8, which are on the “left” side of the line, as shown in the diagram below:



2016 Canadian Computing Olympiad

Day 1, Problem 3

Legends

Time Limit: 2 seconds

Problem Description

The country of Canadia consists of a network of cities and roads. Each road can be traversed in both directions. It is possible to get from any city to any other city using the roads.

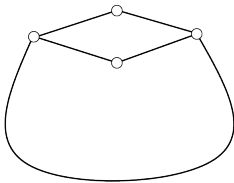

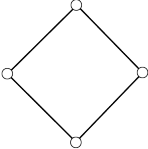
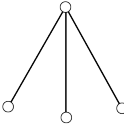
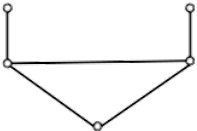
Suzie studies the creation myths of the Canadiaan people. She is particularly interested in five myths (which correspond to the five subtasks of this problem). The myths are very similar. Each myth has the following form:

In the beginning, Canadia's road network had a particular structure. As time went on, the network was modified to meet the needs of Canadia's growing population. Each modification had one of the following forms:

- A road was built between two cities that did not yet have a road going directly between them.
- A new city was built. Cities built in this way were not initially connected to any existing cities.
- A city u grows too large and is split into two cities v and w . The cities originally joined directly to u by a road are partitioned into sets A and B . A road is built from each city in A to v , from each city in B to w and from v to w . For example,



The five myths only differ in the structure that they believe Canadia began with. Here are the original structures, according to each myth:

Subtask 1 [The Myth of the Flask]	Subtask 2 [The Myth of the Moon]
	
Subtask 3 [The Myth of the Sun]	Subtask 4 [The Myth of the Eagle's Talon]
	
Subtask 5 [The Myth of the Fox]	
	

For each subtask, you must take the layout of Canadia as input and determine whether the myth might be correct.

Subtasks are worth 5 marks each.

Input Specification

The first line contains a single integer S ($1 \leq S \leq 5$) representing the subtask which you must solve. The second line contains an integer T ($1 \leq T$) representing the number of test cases. Each test case consists of a blank line, followed by two integers N and M ($2 \leq N$, $1 \leq M$) representing the number of cities and roads, respectively. The cities are numbered from 1 to N . Then M lines follow, each containing two integers a and b ($1 \leq a, b \leq N$) representing two cities connected by a road. No road connects a city to itself. No two roads connect the same pair of cities. It is possible to get from any city to any other city using the roads.

In subtask 3, you may assume that the sum of N over all test cases is at most 10^5 . In all other subtasks, the sum of N over all test cases is at most 1000.

The same condition holds for M . In particular, in subtask 3, you may assume that the sum of M over all test cases is at most 10^5 . In all other subtasks, the sum of M over all test cases is at most 1000.

Output Specification

For each test case, output a single line containing the string YES or the string NO.

Sample Input 1

```

1
2

4 5
1 2
2 3
3 4
1 3
2 4

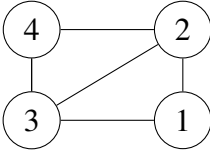
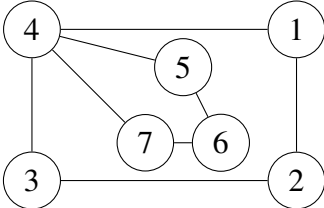
7 8
1 2
2 3
3 4
4 1
4 5
5 6
6 7
7 4
    
```

Output for Sample Input 1

```

YES
NO
    
```

Explanation for Output for Sample Input 1

Test Case Number	Network	Explanation
1		matches The Myth of the Flask
2		cannot match The Myth of the Flask

Sample Input 2

2
2


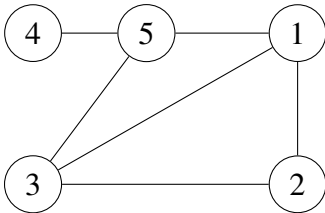
2 1
1 2

5 6
1 3
5 1
2 3
4 5
1 2
3 5

Output for Sample Input 2

NO
YES

Explanation for Output for Sample Input 2

Test Case Number	Network	Explanation
1		cannot match The Myth of the Moon
2		matches The Myth of the Moon, since we can add cities 4 and 5 along with some extra roads to the Moon formed by cities 1, 2 and 3.

Sample Input 3

3

2

7 8

1 2

2 3

3 4

4 1

4 5

5 6

6 7

7 4

8 8

1 2

2 3

3 4

4 5

5 6

6 1

7 3

8 7

Output for Sample Input 3

YES

YES

Explanation of Output for Sample Input 3

Test Case Number	Network	Explanation
1		can match The Myth of the Sun, on cities 1, 2, 3 and 4
2		can match The Myth of the Sun, on cities 1, 2, 3 and 4 where some new cities have been inserted between cities 1 and 4

Sample Input 4

4
2

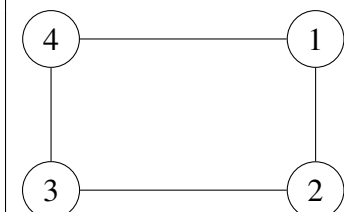
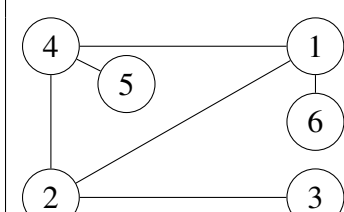
4 4
1 2
2 3
3 4
4 1

6 6
1 2
2 3
1 4
4 5
2 4
1 6

Output for Sample Input 4

NO
YES

Explanation of Output for Sample Input 4

Test Case Number	Network	Explanation
1		cannot match The Myth of the Talon
2		can match The Myth of the Talon on cities 1, 2, 4 and 6

Sample Input 5

5
2

5 5
1 2
2 3
2 4
4 5
3 5

6 6
1 2
2 3
1 4
4 5
2 4
1 6

Output for Sample Input 5

NO
YES

Explanation of Output for Sample Input 5

Test Case Number	Network	Explanation
1	<pre>graph LR; 1 --- 2; 2 --- 4; 2 --- 3; 4 --- 5;</pre>	cannot match The Myth of the Fox
2	<pre>graph LR; 4 --- 1; 4 --- 2; 4 --- 5; 1 --- 2; 1 --- 3; 1 --- 6; 2 --- 3;</pre>	can match The Myth of the Fox, on cities 1, 2, 4, 5 and 6

2016 Canadian Computing Olympiad

Day 2, Problem 1

O Canada

Time Limit: 1 second

Problem Description

In this problem, a *grid* is an N -by- N array of cells, where each cell is either red or white.

Some grids are *similar* to other grids. Grid A is similar to grid B if and only if A can be transformed into B by some sequence of *changes*. A change consists of selecting a 2-by-2 square in the grid and flipping the colour of every cell in the square. (Red cells in the square will become white; white cells in the square will become red.)

You are given G grids. Count the number of pairs of grids which are similar. (Formally, number the grids from 1 to G , then count the number of tuples (i, j) such that $1 \leq i < j \leq G$ and grid i is similar to grid j .)

Input Specification

The first line of input contains N ($2 \leq N \leq 10$), the size of the grids. The second line contains G ($2 \leq G \leq 10000$), the number of grids. The input then consists of $N \cdot G$ lines, where each line contains N characters, where each character is either R or W, indicating the colour (red or white) for that element in the grid. Moreover, after the first two lines of input, the next N lines describe the first grid, the following N lines describe the second grid, and so on.

For 12 out of the 25 marks available for this question, $2 \leq G \leq 10$.

Output Specification

Output the number of pairs of grids which are similar.

Sample Input

```
2
2
RW
WR
WR
RW
```

Output for Sample Input

```
1
```

Explanation for Output for Sample Input

There are exactly two grids, and they are similar because the first grid can be transformed into the second grid using one change (selecting the 2-by-2 square consisting of the entire grid).

2016 Canadian Computing Olympiad

Day 2, Problem 2

Zombie Apocalypse

Time Limit: 2 seconds

Problem Description

Your country has a problem with zombies. That is, it has zombies, which are a problem. Thankfully, you are gainfully employed at the Forsenic Institute for Zoology and Zombie Emerging Studies (FIZZES), and your job is simply to give a measure of how bad the problem is.

You have mapped out your country on an N -by- M array of cells marked with non-negative integers.

You have the exact locations of all the zombies, and know that no two zombies are in the same location. The cells containing a zombie are marked with 0. Next, all the unmarked cells touching a cell (where *touching a cell* means touching on any side or corner of a cell; so each cell touches up to 8 other cells) marked with 0 are marked with 1. Then, all the unmarked cells touching a cell marked with 1 are marked with 2. This process continues until all the cells are marked. These numbers indicate the level of concern your office has about the spread of zombies.

A small example is shown below.

```
2 2 1 1 1 2
2 1 1 0 1 2
2 1 0 1 1 2
2 1 1 1 2 2
2 2 2 2 2 3
```

Your boss has given you an integer Q , and you must determine the number of cells which are marked with the integer Q .

Input Specification

The first line of input will contain two space-separated integers N and M ($1 \leq N \leq 10^9$; $1 \leq M \leq 10^9$) indicating the size of the grid. The next line contains the number K ($1 \leq K \leq 2000$), indicating the number of cells that contain zombies. The next K lines each contain two space-separated integers $r_i c_i$ indicating the row and column of the i th zombie ($1 \leq r_i \leq N$; $1 \leq c_i \leq M$). No two zombies are in the same cell: thus if $i \neq j$ then $(r_i, c_i) \neq (r_j, c_j)$. The last line will contain the integer Q ($0 \leq Q \leq N + M$).

For 5 of the 25 marks available, $N \leq 1000$ and $M \leq 1000$.

For an additional 5 of the 25 marks available, $K \leq 50$.

For an additional 5 of the 25 marks available, $N \leq 1000$.

Output Specification

Output the number of cells in the grid that are marked with the integer Q .

Sample Input

```
5 6
2
3 3
2 4
2
```

Output for Sample Input

```
15
```

Explanation for Output for Sample Input

The sample input is the example shown above, which has 15 2's.

2016 Canadian Computing Olympiad

Day 2, Problem 3

Pirates

Time Limit: 10 seconds

Problem Description

A group of N pirates found K gold coins. They must decide on a way to distribute the coins amongst themselves. They have agreed on the following rules:

The oldest pirate proposes a distribution. (You can assume that all the pirates' ages are distinct.) A distribution assigns a non-negative integer number of coins to each pirate such that the sum of these numbers equals K .

Then, each pirate will vote either 'yes' or 'no' on the proposal. The number of 'yes' votes required for the proposal to pass depends on the number of pirates. If there are X pirates, then $V[X]$ 'yes' votes are required for the proposal to pass. If the proposal passes, the coins are assigned according to the proposed distribution and the process ends. Otherwise, the oldest pirate is thrown overboard and the process is repeated without him.

The pirates act according to the following rules. The rules are given in order of priority; for example, rule 2 is only applied to distinguish between multiple options that are optimal according to rule 1.

1. A pirate will act to prevent himself from being thrown overboard.
2. A pirate will act to maximize the number of coins he receives.
3. A pirate will act to maximize the number of pirates thrown overboard (excepting himself, because rule 1 takes priority).
4. A pirate will act to maximize the number of coins received by the oldest pirate. If there are still multiple choices that fit these rules, he will maximize the gold received by the second-oldest pirate, then the third-oldest pirate, etc.

If there are multiple options that are optimal according to these rules, then the pirate chooses an action arbitrarily. (You can assume that the answer to this problem does not depend on the pirate's choice in this case.) Additionally, all the pirates are perfectly logical and know all the information contained in this problem statement. They cannot form agreements or coalitions because they do not trust each other.

We will number the pirates from 1 to N , where these are numbered from the youngest (pirate 1) to the oldest (pirate N).

If there were only i pirates (where $i = 1, \dots, N$), how many coins would the oldest of them get?

Input Specification

The first line of input will be the number N ($2 \leq N \leq 10^6$).

The second line of input will be the integer K ($1 \leq K \leq 10^{18}$).

The next N lines of input contain $V[i]$ ($1 \leq V[i] \leq i$) indicating the number of ‘yes’ votes required for a proposal to pass if there are i pirates remaining ($i = 1, \dots, N$).

For 5 of the 25 available marks for this problem $N \leq 2000$.

For an additional 5 of the 25 available marks for this problem $\max(1, i - 3) \leq V[i] \leq i$ for all $i = 1, \dots, N$.

For an additional 5 of the 25 available marks for this problem $K = 10^{18}$.

Output Specification

The output should consist of N integers, printed one per line. The i th line of output is the number of coins that the i th pirate would get if they were the oldest pirate; in other words, if only pirates $1, \dots, i$ existed. If the i th pirate is thrown overboard, output -1 on the i th line.

Sample Input 1

```
5
100
1
1
2
2
3
```

Output for Sample Input 1

```
100
100
99
99
98
```

Explanation for Output for Sample Input 1

If there are 2 pirates left, pirate 2 can propose that all of the gold coins go to him. Only 1 vote is required for this proposal to pass, so he can guarantee that it passes by voting for it.

If there are 3 pirates left, pirate 3 needs someone else to vote for his proposal. He can ensure this by giving 1 coin to pirate 1 and 99 to himself. Pirate 1 knows that if the proposal doesn’t pass, he will receive nothing. So a single coin is enough to secure his vote.

If there are 4 pirates left, pirate 4 gives 1 gold coin to pirate 2 and 99 to himself.

If there are 5 pirates left, pirate 5 gives 1 gold coin to pirates 1 and 3 and keeps 98 coins for himself.

Sample Input 2

```
5
100
1
2
3
4
4
```

Output for Sample Input 2

```
100
-1
-1
-1
100
```

Explanation for Output for Sample Input 2

In this case, a full consensus is required for a proposal to pass, except when there are 5 pirates, in which case only 4 of the 5 votes are required.

If there is full consensus required, and there are 2 or more pirates, the youngest pirate will vote against the proposal to maximize their coins and also throw the most pirates overboard.

In the case where there are 5 pirates, the oldest pirate will propose to give himself 100 coins. Everyone except the youngest pirate will vote 'yes', because if this proposal doesn't pass, the youngest pirate will get all of them thrown overboard and then take the coins for himself when only he remains. Since the pirates don't want to be thrown overboard, they will vote 'yes', even though the oldest pirate will offer them nothing.

Sample Input 3

```
4
100
1
1
2
3
```

Output for Sample Input 3

```
100
100
```

99

97

Explanation for Output for Sample Input 3

The first three cases were explained in Sample Input 1.

When there are 4 pirates, the oldest pirate needs 3 votes. He will give 2 coins to the youngest pirate and 1 coin to the second-youngest pirate, keeping the rest for himself.

Sample Input 4

4

2

1

1

2

3

Output for Sample Input 4

2

2

1

-1

Explanation for Output for Sample Input 4

The situation is the same as in Example 3, but now there are only 2 coins. With 1, 2 or 3 pirates, we have the same situations as in Example 3.

With 4 pirates, the oldest pirate does not have enough coins to ensure the 3 votes which he needs, so he will be thrown overboard.