Problem A
Altitude Sickness

Time limit: 3 seconds

Memory limit: 1024 megabytes

Problem Description

Froggy's class plans a hiking for their graduation trip. They decide to stay in the Green Mountains for 2 days. Froggy has a map with n-by-m grids on it. Each grid indicates 1 acre with its height noted.

However, some of Froggy's classmates have altitude sickness. Given each classmate's maximum altitude that they can endure, please calculate how many acres they can stay safely for the trip.

Note: acre is a unit of area.

Input Format

The first line consists of 3 space-separated integers n, m and q. n and m indicate the size of the map, and q is the number of queries.

The following n lines contain m space-separated integers each, the j^{th} element in the i^{th} line a_{ij} is the altitude of the grid cell on the intersection of the i^{th} row and the j^{th} column.

For next q lines, each line is a query with one integer h indicating the maximum height that a Froggy's classmate's maximum altitude that they can endure.

Output Format

For each query, please output the area (in acres) that the classmate can stay safely.

Technical Specification

- $1 \leq n \times m \leq 10^5$
- $1 \leq q \leq 10^5$
- $1 \leq a_{ij} \leq 10^{18}$
- $1 \leq h \leq 10^{18}$

Sample Input 1

```
3 5 4
4 2 5 1 8
1 4 2 8 9
9 9 2 3 5
1
3
6
8
```

Sample Output 1

```
2
6
10
12
```

Problem B
Towers

Time limit: 3 seconds

Memory limit: 1024 megabytes

Problem Description

In Froggy's city, there are n intersections and m roads in a city. Each road connects two different intersections, and there is no more than one road between any pair of intersections.

The mayor plans to build n towers to promote tourism. The heights of the towers are $h_1, h_2, \ldots, h_n$. The towers must be placed at the intersections of roads, and we can only place at most one tower on each intersection.

If we place a tower of height h on an intersection of k roads, then we must build h steps of stairs for the entrance of each road. In total, we have to build kh steps of stairs for such case.

Froggy is the contractor and he wants to minimize the cost of the construction of the stairs. He can decide the placement of the towers without violating the aforementioned constraints. He wonders how many steps of stairs are required to build the towers. Please write a program to compute the minimum number of steps of stairs.

Input Format

The first line contains two integers n and m indicating that there are n intersections of roads and m roads in Froggy's city. The i^{th} of the following m lines contains two integers u_i and v_i indicating that there is a road between the u_i^{th} and the v_i^{th} intersections. The last line contains n integers $h_1, h_2, \ldots, h_n$ indicating the heights of the towers.

Output Format

Output the minimum number of steps of stairs to build the towers.

Technical Specification

- $1 \le n \le 2 \cdot 10^5$
- $0 \le m \le 10^6$
- $1 \le u_i < v_i \le n$, for i in $[1, m]$
- $u_i \ne u_j$ or $v_i \ne v_j$ for i, j in $[1, m]$ and $i \ne j$
- $1 \le h_i \le 10^6$, for i in $[1, n]$

Sample Input 1

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

Sample Output 1

```
10
```

Problem C
Froggy and Froggo

Time limit: 8 seconds

Memory limit: 1024 megabytes

Problem Description

The king of the frogs has two sons, Froggy and Froggo. He wants to divide his kingdom into two parts, one for Froggy and one for Froggo.

The kingdom of the frogs is a square of $n \times n$ grids, with Froggy's castle at the top-left corner $(1, 1)$ and Froggo's castle at the bottom-right corner (n, n). Each grid cell has a value, and these values may be different.

The king wants to be fair, but he doesn't want an over complicated partition. He needs your help to generate a partition satisfying the following restrictions:

- The boundary of the partition is a path.
- The boundary starts at the bottom-left corner of some grid cell lying on the bottom row or the leftmost column of the grid.
- The boundary ends at the top-right corner of some grid cell lying on the top row or the rightmost column of the grid.
- The boundary consists of only edges of grid cells.
- If we walk along the boundary from its start to its end, we only go either upward or right. Moreover, we can only make *at most five* turns.
- The sum of values belongs to Froggy equals the sum of values belongs to Froggo.

Please write a program to calculate the minimum turns of a boundary satisfying all restriction above. If the minimum number of turns is greater than 5, your program should simply output `NO`.

Input Format

The first line contains an integer T indicating the number of test cases. The first line of each testcase contains an integer n. The following n lines contain n integers each, the j^{th} element in the i^{th} line a_{ij} is the value of the grid cell lying on the intersection of the i^{th} row and the j^{th} column.

Output Format

If the minimum number of turns is at most five turns, output `YES` and the minimum number of turns. Otherwise, output `NO`.

Technical Specification

- $1 \le T \le 20$
- $1 \le n \le 50$
- $1 \le a_{ij} \le 10^4$
- $1 \le i, j \le n$
- The sum of all n is at most 50.

Sample Input 1

```
2
3
3 3 2
4 2 2
2 1 1
3
5 5 5
5 1 1
1 1 16
```

Sample Output 1

```
YES 3
YES 2
```

Sample Input 2

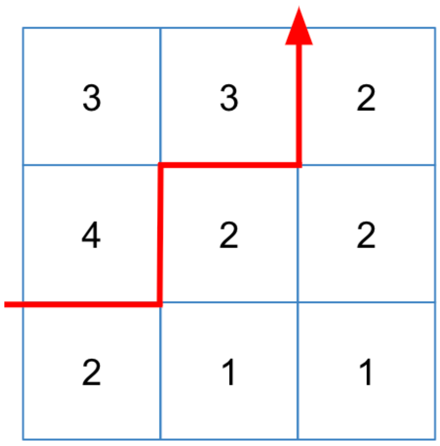
```
1
2
15 3
3 3
```

Sample Output 2

```
NO
```

Note

The following picture is the partition corresponding to the first test case in sample input 1.



Problem D
Robust Eggs

Time limit: 1 second

Memory limit: 1024 megabytes

Problem Description

Hsinchu 1001 is famous for the height of the building. It has 1001 floors, and people use its height to examine the robustness of items. An item has a robustness of k levels if the following conditions hold.

1. The item will be broken when throwing the item from $k + 1$ floor.
2. The item will not be broken when throwing the item from k floor.
3. If throw the item and it is not broken, the robustness will be the same.

We can repeat to examine the robustness as long as the item is not broken.

Froggy is given three eggs with robustness of n levels where n is an unknown integer between 1 to 1000. He wants to know the accurate n, but he has no idea how to find the answer.

Your task is to design a series of tests to find the answer for Froggy. A test is "ask Froggy to throw an egg from m floor and to report whether the egg is broken." for integer m between 1 and 1001. You may operate the new test after all results of previous tests came out. There are two restrictions.

1. Froggy must have at least one egg which is not broken.
2. Froggy cannot throw more than thirty times.

Please tell Froggy the accurate n when you figure out the answer from the tests.

Input Format

This is an interactive problem. The i^{th} line is the result of the i^{th}. The result of a test is either `BROKEN` or `SAFE`. The former means that the egg is broken while the latter is not.

Output Format

The output has $q+1$ lines if you ask Froggy to perform q tests. Each of the first q lines contains a question mark `?` and an integer m. It means that the test is asking Froggy to throw an egg from m floor. The last line contains an exclamation mark `!` and an integer n. It means that the eggs have robustness of n levels.

Technical Specification

- $q \leq 30$
- $1 \leq m \leq 1001$

Sample Input 1

```
BROKEN
BROKEN
SAFE
BROKEN
```

Sample Output 1

```
? 100
? 10
? 5
? 6
! 5
```

Sample Input 2

```
SAFE
BROKEN
```

Sample Output 2

```
? 1000
? 1001
! 1000
```

Note

This is an interactive problem. You must flush the standard output right after you output a newline character. Otherwise, the judge script will not be able to read from your program.

For `C` users: please use the following statement to flush the standard output.

```
fflush(stdout);
```

For `C++` users: please synchronize `ios_base` and `stdio`, and tie `cin` with `cout`. In other words, please do NOT use any of the following "optimization" for I/O.

```
// DO NOT USE THE FOLLOWING STATEMENTS!!
ios_base::sync_with_stdio(false);
cin.tie(0);
```

Moreover, please use `endl` to output a newline character and flush the standard output. For example, you may output in the following style.

```
cout << "? " << floor << endl;
```

For Python users: Please import the `sys` module, and then use `sys.stdout.flush()` to flush the standard output.

For Java users: Please use `System.out.flush()` to flush the standard output.

For Kotlin users: Please use `println` to output with automatic flushing. For example, you may use the following code to print.

```
println("? $floor")
```

Problem E
Rainbow Numbers

Time limit: 1 second

Memory limit: 1024 megabytes

Problem Description

Froggy loves numbers and rainbows. Today, he wants to combine these two of his favorite things. He defines rainbow numbers are numbers satisfying the following constraints.

- The number is a non-negative integer.
- The decimal representation of the number can be written as $a_1 a_2 \cdots a_n$ where $a_i \in \{0, 1, \ldots, 9\}$ for $i \in \{1, 2, \ldots, n\}$.
- There is no leading zeros. That is, $a_1 \neq 0$ if $n > 1$.
- Any two consecutive digits of the number must different. That is, for $1 < i \leq n$, $a_{i-1} \neq a_i$.

For instances, 1213 and 384379 are rainbow numbers. Note that 2334 is not a rainbow number, because there are consecutive 3's. Just like rainbows, two consecutive tracks are in different colors.

Given two integers L and R, please help Froggy find out how many rainbow numbers there are within the interval $[L, R]$.

Input Format

The first line contains an integer T indicating the number of testcases. Each test case is one line containing two integers L and R.

Output Format

For each testcase, output how many rainbow numbers are in the interval $[L, R]$. If there are more than 1000000006, then output the answer modulo 1000000007.

Technical Specification

- $1 \leq T \leq 20$
- $0 \leq L \leq R < 10^{100}$

Sample Input 1

```
1
5 20
```

Sample Output 1

```
15
```

Sample Input 2

```
2
330 339
330 340
```

Sample Output 2

```
0
1
```