A. Cow and Haybales

time limit per test: 2 seconds
memory limit per test: 256 megabytes
input: standard input
output: standard output

The USA Construction Operation (USACO) recently ordered Farmer John to arrange *a row* of n haybale piles on the farm. The i-th pile contains a_i haybales.

However, Farmer John has just left for vacation, leaving Bessie all on her own. Every day, Bessie the naughty cow can choose to move one haybale in any pile to an adjacent pile. Formally, in one day she can choose any two indices i and j ($1 \le i, j \le n$) such that $|i-j| = 1$ and $a_i > 0$ and apply $a_i = a_i - 1$, $a_j = a_j + 1$. She may also decide to not do anything on some days because she is lazy.

Bessie wants to maximize the number of haybales in pile 1 (i.e. to maximize a_1), and she only has d days to do so before Farmer John returns. Help her find the maximum number of haybales that may be in pile 1 if she acts optimally!

Input

The input consists of multiple test cases. The first line contains an integer t ($1 \le t \le 100$) — the number of test cases. Next $2t$ lines contain a description of test cases — two lines per test case.

The first line of each test case contains integers n and d ($1 \le n, d \le 100$) — the number of haybale piles and the number of days, respectively.

The second line of each test case contains n integers $a_1, a_2, \ldots, a_n$ ($0 \le a_i \le 100$) — the number of haybales in each pile.

Output

For each test case, output one integer: the maximum number of haybales that may be in pile 1 after d days if Bessie acts optimally.

Example

input

```
3
4 5
1 0 3 2
2 2
100 1
1 8
0
```

output

```
3
101
0
```

Note

In the first test case of the sample, this is one possible way Bessie can end up with 3 haybales in pile 1:

- On day one, move a haybale from pile 3 to pile 2
- On day two, move a haybale from pile 3 to pile 2
- On day three, move a haybale from pile 2 to pile 1
- On day four, move a haybale from pile 2 to pile 1
- On day five, do nothing

In the second test case of the sample, Bessie can do nothing on the first day and move a haybale from pile 2 to pile 1 on the second day.

B. Cow and Friend

time limit per test: 2 seconds
memory limit per test: 256 megabytes
input: standard input
output: standard output

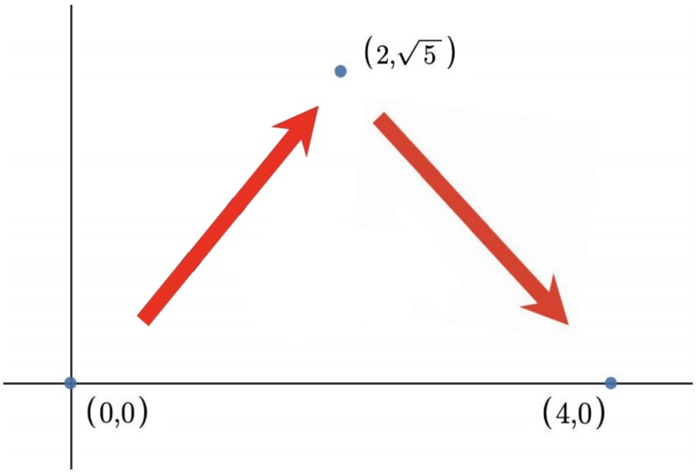
Bessie has way too many friends because she is everyone's favorite cow! Her new friend Rabbit is trying to hop over so they can play!

More specifically, he wants to get from $(0,0)$ to $(x,0)$ by making multiple hops. He is only willing to hop from one point to another point on the 2D plane if the Euclidean distance between the endpoints of a hop is one of its n favorite numbers: $a_1, a_2, \ldots, a_n$.

What is the minimum number of hops Rabbit needs to get from $(0,0)$ to $(x,0)$? Rabbit may land on points with non-integer coordinates. It can be proved that Rabbit can always reach his destination.

Recall that the Euclidean distance between points (x_i, y_i) and (x_j, y_j) is $\sqrt{(x_i - x_j)^2 + (y_i - y_j)^2}$.

For example, if Rabbit has favorite numbers 1 and 3 he could hop from $(0,0)$ to $(4,0)$ in two hops as shown below. Note that there also exists other valid ways to hop to $(4,0)$ in 2 hops (e.g. $(0,0) \to (2, -\sqrt{5}) \to (4,0)$).



Here is a graphic for the first example. Both hops have distance 3, one of Rabbit's favorite numbers.

In other words, each time Rabbit chooses some number a_i and hops with distance equal to a_i in any direction he wants. The same number can be used multiple times.

Input

The input consists of multiple test cases. The first line contains an integer t $(1 \le t \le 1000)$ — the number of test cases. Next $2t$ lines contain test cases — two lines per test case.

The first line of each test case contains two integers n and x $(1 \le n \le 10^5, 1 \le x \le 10^9)$ — the number of favorite numbers and the distance Rabbit wants to travel, respectively.

The second line of each test case contains n integers $a_1, a_2, \ldots, a_n$ $(1 \le a_i \le 10^9)$ — Rabbit's favorite numbers. It is guaranteed that the favorite numbers are distinct.

It is guaranteed that the sum of n over all the test cases will not exceed 10^5.

Output

For each test case, print a single integer — the minimum number of hops needed.

Example

input

```
4
2 4
1 3
3 12
3 4 5
1 5
5
2 10
15 4
```

output

```
2
3
1
2
```

Note

The first test case of the sample is shown in the picture above. Rabbit can hop to $(2, \sqrt{5})$, then to $(4,0)$ for a total of two hops. Each hop has a distance of 3, which is one of his favorite numbers.

In the second test case of the sample, one way for Rabbit to hop 3 times is: $(0,0) \to (4,0) \to (8,0) \to (12,0)$.

In the third test case of the sample, Rabbit can hop from $(0,0)$ to $(5,0)$.

In the fourth test case of the sample, Rabbit can hop: $(0,0) \to (5, 10\sqrt{2}) \to (10,0)$.

C. Cow and Message

time limit per test: 2 seconds

memory limit per test: 256 megabytes

input: standard input
output: standard output

Bessie the cow has just intercepted a text that Farmer John sent to Burger Queen! However, Bessie is sure that there is a secret message hidden inside.

The text is a string s of lowercase Latin letters. She considers a string t as hidden in string s if t exists as a subsequence of s whose indices form an arithmetic progression. For example, the string aab is hidden in string aaabb because it occurs at indices 1, 3, and 5, which form an arithmetic progression with a common difference of 2. Bessie thinks that any hidden string that occurs the most times is the secret message. Two occurrences of a subsequence of S are distinct if the sets of indices are different. Help her find the number of occurrences of the secret message!

For example, in the string aaabb, a is hidden 3 times, b is hidden 2 times, ab is hidden 6 times, aa is hidden 3 times, bb is hidden 1 time, aab is hidden 2 times, aaa is hidden 1 time, abb is hidden 1 time, aaab is hidden 1 time, aabb is hidden 1 time, and aaabb is hidden 1 time. The number of occurrences of the secret message is 6.

Input

The first line contains a string s of lowercase Latin letters ($1 \le |s| \le 10^5$) — the text that Bessie intercepted.

Output

Output a single integer — the number of occurrences of the secret message.

Examples

input
```
aaabb
```
output
```
6
```

input
```
usaco
```
output
```
1
```

input
```
lol
```
output
```
2
```

Note

In the first example, these are all the hidden strings and their indice sets:

- a occurs at (1), (2), (3)
- b occurs at (4), (5)
- ab occurs at $(1,4)$, $(1,5)$, $(2,4)$, $(2,5)$, $(3,4)$, $(3,5)$
- aa occurs at $(1,2)$, $(1,3)$, $(2,3)$
- bb occurs at $(4,5)$
- aab occurs at $(1,3,5)$, $(2,3,4)$
- aaa occurs at $(1,2,3)$
- abb occurs at $(3,4,5)$
- aaab occurs at $(1,2,3,4)$
- aabb occurs at $(2,3,4,5)$
- aaabb occurs at $(1,2,3,4,5)$

Note that all the sets of indices are arithmetic progressions.
In the second example, no hidden string occurs more than once.

In the third example, the hidden string is the letter l.

D. Cow and Fields

time limit per test: 2 seconds
memory limit per test: 256 megabytes
input: standard input
output: standard output

Bessie is out grazing on the farm, which consists of n fields connected by m bidirectional roads. She is currently at field 1, and will return to her home at field n at the end of the day.

The Cowfederation of Barns has ordered Farmer John to install one extra bidirectional road. The farm has k special fields and he has decided to install the road between two different special fields. He may add the road between two special fields that already had a road directly connecting them.

After the road is added, Bessie will return home on the shortest path from field 1 to field n. Since Bessie needs more exercise, Farmer John must **maximize** the length of this shortest path. Help him!

Input

The first line contains integers n, m, and k ($2 \le n \le 2 \cdot 10^5$, $n-1 \le m \le 2 \cdot 10^5$, $2 \le k \le n$) — the number of fields on the farm, the number of roads, and the number of special fields.

The second line contains k integers $a_1, a_2, \ldots, a_k$ ($1 \le a_i \le n$) — the special fields. All a_i are distinct.

The i-th of the following m lines contains integers x_i and y_i ($1 \le x_i, y_i \le n$, $x_i \ne y_i$), representing a bidirectional road between fields x_i and y_i.

It is guaranteed that one can reach any field from every other field. It is also guaranteed that for any pair of fields there is at most one road connecting them.

Output

Output one integer, the **maximum** possible length of the shortest path from field 1 to n after Farmer John installs one road optimally.

Examples

input

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

output

```
3
```

input

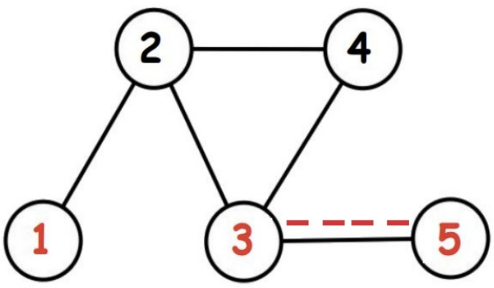
```
5 4 2
2 4
1 2
2 3
3 4
4 5
```

output

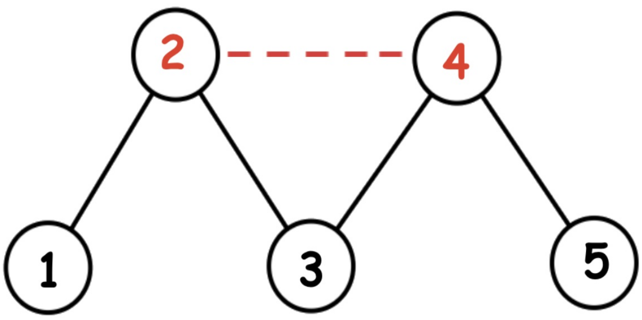
```
3
```

Note

The graph for the first example is shown below. The special fields are denoted by red. It is optimal for Farmer John to add a road between fields 3 and 5, and the resulting shortest path from 1 to 5 is length 3.



The graph for the second example is shown below. Farmer John must add a road between fields 2 and 4, and the resulting shortest path from 1 to 5 is length 3.



E. Cow and Treats

time limit per test: 1 second
memory limit per test: 256 megabytes
input: standard input
output: standard output

After a successful year of milk production, Farmer John is rewarding his cows with their favorite treat: tasty grass!

On the field, there is a row of n units of grass, each with a sweetness s_i. Farmer John has m cows, each with a favorite sweetness f_i and a hunger value h_i. He would like to pick two disjoint subsets of cows to line up on the left and right side of the grass row. There is no restriction on how many cows must be on either side. The cows will be treated in the following manner:

- The cows from the left and right side will take turns feeding in an order decided by Farmer John.
- When a cow feeds, it walks towards the other end without changing direction and eats grass of its favorite sweetness until it eats h_i units.
- The moment a cow eats h_i units, it will fall asleep there, preventing further cows from passing it from both directions.
- If it encounters another sleeping cow or reaches the end of the grass row, it will get upset. Farmer John absolutely does not want any cows to get upset.

Note that grass does not grow back. Also, to prevent cows from getting upset, not every cow has to feed since FJ can choose a subset of them.

Surprisingly, FJ has determined that sleeping cows are the most satisfied. If FJ orders optimally, what is the maximum number of sleeping cows that can result, and how many ways can FJ choose the subset of cows on the left and right side to achieve that maximum number of sleeping cows (modulo $10^9 + 7$)? The order in which FJ sends the cows does not matter as long as no cows get upset.

Input

The first line contains two integers n and m ($1 \le n \le 5000$, $1 \le m \le 5000$) — the number of units of grass and the number of cows.

The second line contains n integers $s_1, s_2, \ldots, s_n$ ($1 \le s_i \le n$) — the sweetness values of the grass.

The i-th of the following m lines contains two integers f_i and h_i ($1 \le f_i, h_i \le n$) — the favorite sweetness and hunger value of the i-th cow. **No two cows have the same hunger and favorite sweetness simultaneously**.

Output

Output two integers — the maximum number of sleeping cows that can result and the number of ways modulo $10^9 + 7$.

Examples

input
```
5 2
1 1 1 1 1
1 2
1 3
```
output
```
2 2
```

input
```
5 2
1 1 1 1 1
1 2
1 4
```
output
```
1 4
```

input
```
3 2
2 3 2
3 1
2 1
```
output
```
2 4
```

input
```
5 1
1 1 1 1 1
2 5
```
output
```
0 1
```

Note

In the first example, FJ can line up the cows as follows to achieve 2 sleeping cows:

- Cow 1 is lined up on the left side and cow 2 is lined up on the right side.
- Cow 2 is lined up on the left side and cow 1 is lined up on the right side.

In the second example, FJ can line up the cows as follows to achieve 1 sleeping cow:

- Cow 1 is lined up on the left side.
- Cow 2 is lined up on the left side.
- Cow 1 is lined up on the right side.
- Cow 2 is lined up on the right side.

In the third example, FJ can line up the cows as follows to achieve 2 sleeping cows:

- Cow 1 and 2 are lined up on the left side.
- Cow 1 and 2 are lined up on the right side.
- Cow 1 is lined up on the left side and cow 2 is lined up on the right side.
- Cow 1 is lined up on the right side and cow 2 is lined up on the left side.

In the fourth example, FJ cannot end up with any sleeping cows, so there will be no cows lined up on either side.

F. Cow and Vacation

time limit per test: 2 seconds
memory limit per test: 256 megabytes
input: standard input
output: standard output

Bessie is planning a vacation! In Cow-lifornia, there are n cities, with $n-1$ bidirectional roads connecting them. It is guaranteed that one can reach any city from any other city.

Bessie is considering v possible vacation plans, with the i-th one consisting of a start city a_i and destination city b_i.

It is known that only r of the cities have rest stops. Bessie gets tired easily, and cannot travel across more than k consecutive roads without resting. In fact, she is so desperate to rest that she may travel through the same city multiple times in order to do so.

For each of the vacation plans, does there exist a way for Bessie to travel from the starting city to the destination city?

Input

The first line contains three integers n, k, and r ($2 \le n \le 2 \cdot 10^5$, $1 \le k, r \le n$) — the number of cities, the maximum number of roads Bessie is willing to travel through in a row without resting, and the number of rest stops.

Each of the following $n-1$ lines contain two integers x_i and y_i ($1 \le x_i, y_i \le n$, $x_i \neq y_i$), meaning city x_i and city y_i are connected by a road.

The next line contains r integers separated by spaces — the cities with rest stops. Each city will appear at most once.

The next line contains v ($1 \le v \le 2 \cdot 10^5$) — the number of vacation plans.

Each of the following v lines contain two integers a_i and b_i ($1 \le a_i, b_i \le n$, $a_i \ne b_i$) — the start and end city of the vacation plan.

Output

If Bessie can reach her destination without traveling across more than k roads without resting for the i-th vacation plan, print YES. Otherwise, print NO.

Examples

input

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

output

```
YES
YES
NO
```

input

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

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

output

```
YES
NO
```

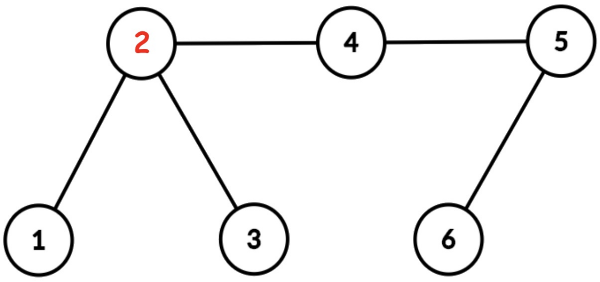
Note

The graph for the first example is shown below. The rest stop is denoted by red.

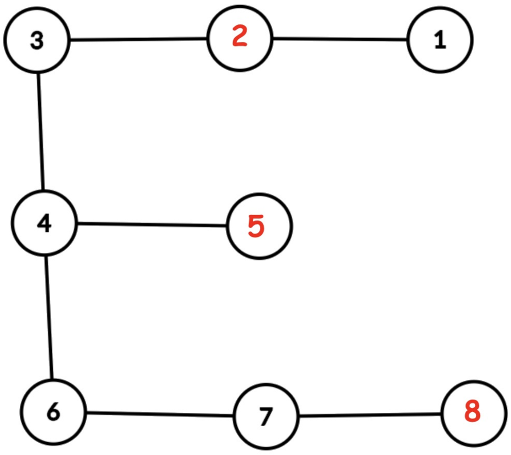
For the first query, Bessie can visit these cities in order: $1, 2, 3$.

For the second query, Bessie can visit these cities in order: $3, 2, 4, 5$.

For the third query, Bessie cannot travel to her destination. For example, if she attempts to travel this way: $3, 2, 4, 5, 6$, she travels on more than 2 roads without resting.



The graph for the second example is shown below.



G. Cow and Exercise

time limit per test: 1 second
memory limit per test: 256 megabytes
input: standard input
output: standard output

Farmer John is obsessed with making Bessie exercise more!

Bessie is out grazing on the farm, which consists of n fields connected by m directed roads. Each road takes some time w_i to cross. She is currently at field 1 and will return to her home at field n at the end of the day.

Farmer John has plans to increase the time it takes to cross certain roads. He can increase the time it takes to cross each road by a nonnegative amount, but the total increase cannot exceed x_i for the i-th plan.

Determine the maximum he can make the shortest path from 1 to n for each of the q independent plans.

Input

The first line contains integers n and m ($2 \le n \le 50$, $1 \le m \le n \cdot (n-1)$) — the number of fields and number of roads, respectively.

Each of the following m lines contains 3 integers, u_i, v_i, and w_i ($1 \le u_i, v_i \le n$, $1 \le w_i \le 10^6$), meaning there is an road from field u_i to field v_i that takes w_i time to cross.

It is guaranteed that there exists a way to get to field n from field 1. It is guaranteed that the graph does not contain self-loops or parallel edges. It is possible to have a road from u to v and a road from v to u.

The next line contains a single integer q ($1 \le q \le 10^5$), the number of plans.

Each of the following q lines contains a single integer x_i, the query ($0 \le x_i \le 10^5$).

Output

For each query, output the maximum Farmer John can make the shortest path if the total increase does not exceed x_i.

Your answer is considered correct if its absolute or relative error does not exceed 10^{-6}.

Formally, let your answer be a, and the jury's answer be b. Your answer is accepted if and only if $\frac{|a-b|}{\max{(1,|b|)}} \le 10^{-6}$.

Example

input

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

output

```
3.0000000000
4.0000000000
4.5000000000
5.0000000000
5.5000000000
```