Task: DEK
Acyclic decomposition

Maratona de Verão, Day 4. Source file `dek.*` Available memory: 32 MB. *23.01.2014*

Byteman is studying directed graphs. He especially likes graphs which do not contain cycles, since this is a class of graphs in which many problems can be solved easily and effectively. Now he is trying to find a method of representing any directed graph as a sum of acyclic graphs.

For a given directed graph he is trying to find a way to divide the set of its edges into a minimal number of subsets in such a way that the graphs constructed using the respective subsets of edges do not contain cycles. Could you write a program which solves Byteman's problem?

Input

The first line of the standard input contains two integers n and m ($1 \leq n, m \leq 100\,000$), denoting the number of vertices and the number of edges in the graph, respectively. The vertices are numbered from 1 to n. Each of the following m lines contains a description of one edge of the graph as a pair of integers a_i, b_i ($1 \leq a_i, b_i \leq n, a_i \neq b_i$). Such a pair denotes a directed edge of the graph from the vertex a_i to the vertex b_i. You may assume that the graph does not contain multiple edges.

Output

The first line of the standard output should contain a single integer k — the minimal number of acyclic graphs in any decomposition of the graph. Each of the following k lines should contain a description of one element of the decomposition, starting with an integer l_i denoting the number of edges in the ith element. It should be followed by an increasing sequence of l_i numbers of edges belonging to the ith element of the decomposition. The edges are numbered from 1 to m in the order in which they appear in the input. Each edge should be present in exactly one element of the decomposition.

If there are multiple correct solutions, your program should output any one of them.

Example

For the input data:

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

the correct result is:

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

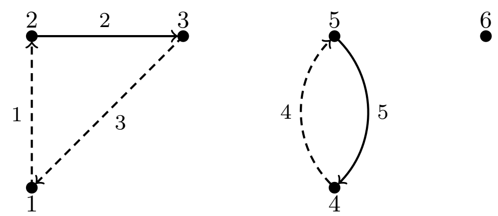


Figure 1: Illustration of the example from the task statement. The circles represent the vertices, while the lines and arcs (continuous and dashed) represent the edges of the graph. The numbers next to the circles are the numbers of the vertices, and the numbers next to the lines/arcs are the numbers of edges. This graph can be decomposed into two acyclic graphs: the edges of the first one are denoted by continuous lines/arcs and the edges of the second one — by dashed lines/arcs.

Task: DZI
Divisors

Maratona de Verão, Day 4. Source file `dzi.*` Available memory: 64 MB. *23.01.2014*

In this problem we are interested in the divisors of a given natural number n. Let us denote the set of all these divisors by $D(n)$. We are given an expression which consists of constant numbers from the set $D(n)$, variables which can assume any values from the set $D(n)$ and binary functions computing the greatest common divisor and the least common multiple.

For a given expression we would like to find out whether its value is constant regardless of the values of all variables.

Input

The first line of the standard input contains one integer t ($1 \leq t \leq 1\,000$) denoting the number of test cases. Each of the following t lines contains a description of a single test case. Each such description starts with an integer n_i ($1 \leq n_i \leq 10^{18}$). It is followed by a description of the expression. An expression is a constant, a variable or a function.

Each number in the description represents a constant. All these numbers are positive divisors of n_i.

Variables are represented as sequences of at most 5 lowercase letters of the English alphabet. Variables represented by the same sequences of letters are considered the same.

The sequences of letters `NWD` and `NWW` represent the functions computing the greatest common divisor and the least common multiple, respectively. The name of a function is followed by a single space, followed by space-separated descriptions of its arguments, which are expressions (hence, the description of the expression is recursive).

You may assume that the total size of the input file does not exceed 2 MB.

Output

Your program should output t lines to the standard output containing the answers to subsequent test cases. The answer to a single test case is one word: `TAK` (meaning *yes* in Polish) or `NIE` (meaning *no* in Polish) stating whether the expression in the test case represents a constant function.

Example

For the input data:

```
3
24 NWD 3 NWW x 12
15 NWD 15 nwd
10 10
```

the correct result is:

```
TAK
NIE
TAK
```

Task: LUN
Telescope

Maratona de Verão, Day 4. Source file `lun.*` Available memory: 64 MB. *23.01.2014*

Byteasar is going to watch a meteor shower tonight. He has read an exact prognosis of its route. He knows that the shower will be composed of n meteors and the i-th meteor will appear at t_i-th second after midnight, i.e., it will be visible between moments $t_i - 1$ and t_i. For each meteor, Byteasar is only interested in watching it the whole time it is visible.

Byteasar will watch the sky from a nearby hill. He is not going to do it with a naked eye. On the hill there is a telescope which perfectly suits an amateur's needs. However, it is not for free. The telescope accepts coins. For each c inserted bythalers it allows to watch the sky for exactly c seconds. Unfortunately the telescope is a bit shabby, so after an insertion of a coin one must wait for r seconds before anything is visible. Every time only a single coin is accepted. Thus, if you insert a c coin, the next coin will be accepted only after at least $r + c$ seconds.

Byteasar has m coins in his pocket with the values of $c_1, \ldots, c_m$ bythalers. He is going to use them to pay for the telescope. He wonders what is the maximum number of meteors he can observe with the telescope.

Input

The first line of the input contains three integers n, m and r ($1 \le n \le 100$, $1 \le m \le 10$, $1 \le r \le 10^8$). The second line is composed of n integers $t_1, \ldots, t_n$ given in the increasing order ($1 \le t_i \le 10^8$). The third line contains m integers $c_1, \ldots, c_m$ ($1 \le c_i \le 10^8$).

Output

The only line of the output should contain one integer, i.e., the maximum number of meteors that Byteasar can observe with the telescope.

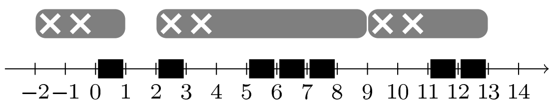
Example

For the input data:

```
7 3 2
1 3 6 7 8 12 13
2 5 1
```

the correct result is:

```
6
```



Explanation of the example: In the figure above black rectangles show seconds during which a meteor is visible. If Byteasar wants to see 6 meteors, he should insert his coins in moments -2, 2 and 9 in the following order of values: 1, 5 and 2 bythalers.

Task: MEC
Matches

Maratona de Verão, Day 4. Source file `mec.*` Available memory: 128 MB. *23.01.2014*

On a Saturday afternoon, n children plan to meet on the football pitch of *Byties* football club. Luckily, the number of children is even and thus they can easily split into two teams and play football.

Byteasar is the club's coach, who is responsible for selecting the players for each game. He know that the children love to compete with each other, so the decided to split the children into teams in such a way that each two children can play against each other in some game (that is, play in opposing teams at most once).

Having considered the skills of the children, Byteasar proposed the line-ups for the following m games. In every match all children will be divided into two teams of $n/2$ players. Help Byteasar determine if every pair of children will play against each other in at least one game.

Input

The first line of the input contains two integers n and m. ($4 \leq n \leq 40\,000$, $1 \leq m \leq 50$) denoting the number of children and the number of games to be played. Each child has a number on its shirt – an integer from 1 to n. These numbers are distinct.

Each of the following m lines contains n pairwise distinct integers from range 1 to n. They denote the line-ups from consecutive games. The first $n/2$ numbers in every line are the numbers of players in the first team, whereas the following $n/2$ numbers specify the numbers of players from the second team.

Output

Your program should output a single word `TAK` (Polish for *yes*) or `NIE` (Polish for *no*), depending on whether each pair of children will play against each other in at least one game, or not.

For

For the input data:

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

the correct result is:

```
TAK
```

whereas for the input data:

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

the correct result is:

```
NIE
```

Task: PAL
Fuel

Maratona de Verão, Day 4. Source file `pal.*` Available memory: 128 MB. *23.01.2014*

In the good old days, all n towns in Byteland were connected by a dense network of two-way roads. The king of Byteland decided to reduce the number of roads, and asked his Chief Computer Scientist for advice. Now, as a result of this decision, Byteland has only $n - 1$ two-way roads connecting pairs of towns in such a way that there is exactly one route between any two towns in Byteland. All the roads are of the same length.

Equipped with a car whose tank has exactly enough fuel to drive on m roads in Byteland, Byteasar has decided to organize a trip that maximizes the number of visited towns. He can start his trip in any of the towns in Byteland, and, similarly, his trip can end anywhere—not necessarily back at the starting town. In maximizing the number of visited towns, Byteasar is allowed to drive multiple times on the same road, in the same or the reverse direction. Your task is to help Byteasar by finding the maximum number of towns that can be visited on one full tank of fuel.

Input

The first line of the input contains two integers, n and m ($2 \leq n \leq 500\,000$, $1 \leq m \leq 200\,000\,000$), where n is the number of towns in Byteland (each numbered uniquely from $\{1, \ldots, n\}$), and m is the number of roads that can be traveled on one tank of fuel.

The next $n - 1$ lines describe Byteland's road network. Each of these lines contains two integers, a and b ($1 \leq a, b \leq n$), indicating that towns a and b are connected with a two-way road.

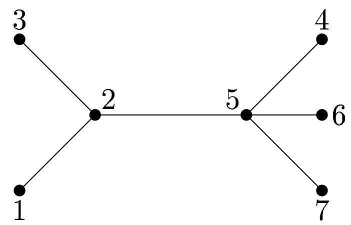
Output

The output consists of one line containing exactly one integer: the maximum number of towns that can be visited on one full tank of fuel.

Example

For the input data:

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



the correct result is:

```
5
```

Explanation of the example: Byteasar can visit a maximum of five different towns. There are several different routes that visit five towns for this example input, including $4 \to 5 \to 7 \to 5 \to 6 \to 5 \to 2$ and $3 \to 2 \to 1 \to 2 \to 5 \to 6 \to 5$.

Task: POD
Coprime Subsets

Maratona de Verão, Day 4. Source file `pod.*` Available memory: 64 MB. *23.01.2014*

How many subsets of the set $\{1, 2, \ldots, n\}$ are coprime? A set of integers is called coprime if every two of its elements are coprime. Two integers are coprime if their greatest common divisor equals 1.

Input

The first line of input contains two integers n and m ($1 \leq n \leq 3000$, $1 \leq m \leq 10^9 + 10$).

Output

Output the number of coprime subsets of $\{1, 2, \ldots, n\}$ modulo m.

Example

For the input data:

```
4 7
```

the correct result is:

```
5
```

Explanation: There are 12 coprime subsets of $\{1, 2, 3, 4\}$: $\emptyset$, $\{1\}$, $\{2\}$, $\{3\}$, $\{4\}$, $\{1, 2\}$, $\{1, 3\}$, $\{1, 4\}$, $\{2, 3\}$, $\{3, 4\}$, $\{1, 2, 3\}$, $\{1, 3, 4\}$. 12 modulo 7 equals 5.

Task: ROZ
Completely different words

Maratona de Verão, Day 4. Source file `roz.*` Available memory: 128 MB. *23.01.2014*

Two strings *completely different* if they differ in every position. You are given a list of n strings of exactly five characters. Write a program that finds pairs of completely different strings in the list.

Input

The first line of input contains a single integer n ($2 \le n \le 50\,000$) denoting the number of strings. Each of the following n lines contains a string of five characters, whose ASCII codes are in the range $[48, 122]$.

Output

The first line of input should contain a single nonnegative integer m denoting the number of (unordered) pairs of completely different strings. Each of the following m lines should contain two integers denoting the numbers of completely different strings. The strings are numbered from 1 to n in the order they are given in the input.

If there are more than 100 000 pairs of completely different strings, output only 100 000 pairs (the pairs may be chosen arbitrarily). In such case, output 100 000 in the first line of the output.

Example

For the input data:

```
3
aB;Va
xBx@a
zc:ng
```

the correct result is:

```
2
1 3
2 3
```

Task: WCZ
Holidays

Maratona de Verão, Day 4. Source file `wcz.*` Available memory: 128 MB. *23.01.2014*

The traffic congestion in Byteotinan Interplanetary Space has increased considerably in recent years. It has become so complex to coordinate the spaceships moving in all possible directions, that the senate of Byteotia has ordered every vessel to move parallel to the axes of a three-dimensional coordinate system.

This has made is difficult for travel agencies, who have started selling *Holidays at the end of the Universe* packages, that is holidays on the most distant planet in the galaxy. Now, the change introduced by the senate means that way of measuring distance has changed.

You have been hired by a travel agency that gave you the coordinates of all planets in the galaxy. Your task is to compute, for each planet, the distance to the furthest planet in the galaxy. The distance between planets at coordinates (x_1, y_1, z_1) and (x_2, y_2, z_2) is given by

$$|x_1 - x_2| + |y_1 - y_2| + |z_1 - z_2|.$$

Input

The first line contains a single integer n ($1 \leq n \leq 500\,000$) specifying the number of planets in the galaxy. Each of the following n lines contains the coordinates of one planet: three integers x_i, y_i, z_i ($-10^9 \leq x_i, y_i, z_i \leq 10^9$). The coordinates of one planet may appear in the input multiple times.

Output

Output n lines. The i-th of these lines should contain a single integer equal to the distance from the planet given in the $i+1$-th line of input to the furthest planet in the galaxy.

Example

For the input data:

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

the correct result is:

```
8
6
6
8
```

Task: ZAJ
Squared words

Maratona de Verão, Day 4. Source file `zaj.*` Available memory: 32 MB. *23.01.2014*

A squared word is a word of the form xx, i.e., consisting of a sequence of letters repeated twice. Some examples of squared words in English are the words: *couscous*, *murmur* (= a low, continuous sound), *tartar* (= hardened dental plaque) and *hotshots*. We would like to find out, for a given word, how many letters should be removed from the word for it to become a squared word (however, not necessarily a correct English word).

Input

The first line of the standard input contains an integer n ($1 \leq n \leq 1000$), denoting the length of the word. The second line contains a word consisting of n lower case English letters.

Output

The first and only line of the standard output should contain a single integer — the minimal number of letters that should be removed from the word for it to become a squared word. We assume that the empty word is a correct squared word.

Example

For the input data:

```
15
tachystoskopach
```

the correct result is:

```
7
```

Explanation of the example: If we remove the letters `y`, `s`, `o`, `s`, `k`, `o` and `p`, we obtain a squared word `tachtach`. In Polish, this word represents a (not really well known) kind of a coach.

Task: ZAM
Surprise

Maratona de Verão, Day 4. Source file `zam.*` Available memory: 128 MB. *23.01.2014*

It's so nice to sleep under the sky and to be woken up by singing birds and the first rays of sun raising from the horizon… Wait, you don't remember going to sleep in the middle of the forest.

What you do remember is that you were covering something with three rectangles. Unsure about what it was, you decide to solve the following problem.

For a given matrix of integers, find k non-overlapping submatrices (in other words, rectangular fragments), such that the sum of numbers in them is as big as possible.

Input

The first line contains three integers n, m i k ($1 \le k \le 3$, $k \le n, m \le 300$) denoting the height and width of the matrix, and the number of submatrices, respectively. The following n lines describe the matrix, row by row. Each of them contains a sequence of m integers a_{ij} ($-20\,000 \le a_{ij} \le 20\,000$).

Output

Output one integer to the standard output equal to the biggest possible sum of integers in k submatrices of the given matrix. The submatrices may touch, but they cannot have any common elements.

Example

For the input data:

```
4 5 2
6 -10 0 3 -6
-8 8 1 -5 3
-7 -3 2 4 -4
2 0 -1 3 -3
```

the correct result is:

```
17
```