Задача A. Assistants

Имя входного файла:	`standard input`
Имя выходного файла:	`standard output`
Ограничение по времени:	`10 seconds`
Ограничение по памяти:	512 mebibytes

Professor Byteasar manages a large research project. In order to finish the project, Byteasar must execute n independent research tasks (numbered by integers from 1 to n).

Every research task can be done by Byteasar or can be delegated to some assistant. If Byteasar decides to do the i-th task by himself, he will need p_i consecutive days to complete it; during these days he cannot do any other tasks. On the other hand, if Byteasar decides to delegate the i-th task to an assistant, he will spend one day to find an adequate one. The engaged assistant will begin working the next day and he or she will use consecutive a_i days to complete the task. Byteasar cannot do any other tasks in the day he is looking for an assistant, but in the following days (when engaged assistant works) he can do something else. You can assume that Byteasar will always find a qualified assistant, no matter how many assistants already work for him.

Professor Byteasar needs your help in planning the following days in such a way that the research project will be finished as soon as possible.

Формат входного файла

The first line of the input contains one integer n ($1 \le n \le 500\,000$), specifying the number of tasks in Byteasar's research project. The i-th of the following n lines contains two integers p_i and a_i ($1 \le p_i \le a_i \le 10^9$), specifying the number of days which are needed for completing the i-th task by Byteasar and his assistant, respectively.

Формат выходного файла

The first line of the output should contain one integer k: the minimum number of days which are needed to finish the research project. The following n lines should contain a schedule which results in completing the project in k days. The i-th of these lines should contain a letter "`B`" or "`A`" and an integer t_i ($1 \le t_i \le k$) separated by a single space: outputting the letter "`B`" denotes that Byteasar should perform the i-th task starting in day t_i (and finishing in day $t_i + p_i - 1$), and outputting the letter "`A`" denotes that Byteasar should devote day t_i to finding an assistant, who will perform the i-th task (in days from $t_i + 1$ to $t_i + a_i$). If there is more than one valid schedule, you can output any of them.

Примеры

standard input	standard output

5
2 6
2 6
2 5
2 2
1 2
	7
	A 1
	B 3
	A 2
	A 5
	B 7

1 1 1	1 B 1

Задача B. Kingdom Division

Имя входного файла:	`standard input`
Имя выходного файла:	`standard output`
Ограничение по времени:	`2 seconds`
Ограничение по памяти:	512 mebibytes

King Byteasar has ruled the Kingdom of Byteotia for several years. Nowadays he is not as strong and healthy as he used to be in the old days. That is why he decided to divide the kingdom into parts that he would give to his sons, the princes of Byteotia. Byteasar has a really large number of sons.

To avoid arguments about succession, Byteasar would like each of the parts to have different size. Also, due to their lack of experience in ruling, a number of youngest princes may not be assigned any part of the kingdom.

The Kingdom of Byteotia can be modeled as an $n \times m$ rectangle divided into $n \cdot m$ unit square fields. Each prince's part must be composed only of whole unit square fields and must be connected, that is, between any two unit square fields in the part there must exist a path composed of unit square fields in which every two consecutive fields are adjacent along an edge. Each unit square field must be assigned to exactly one prince's part.

Byteasar would like as many princes to be assigned non-empty parts of the kingdom as possible. Help him perform such a division.

Формат входного файла

The input contains two integers n and m ($1 \le n, m \le 1\,000$), which specify the size of the rectangle.

Формат выходного файла

The first line of the output should contain one integer k: the maximum number of princes that can be assigned a part of the kingdom. This number should be followed by an example description of a division: n lines containing m characters from the set $\{\mathtt{A}, \ldots, \mathtt{Z}\}$ each. The parts of the kingdom are represented by maximal (that is, non-extendible in either side) connected parts composed of the same letter. Note that different parts may still be described by the same letter. The size of each part (that is, the number of letters in the part) must be different.

It is guaranteed that, for every possible answer, there exists a way to denote its parts by letters in the above format using no more than 26 letters.

Примеры

standard input	standard output
4 5	5 ABCCE BBCCE DDDEE DDEEE
1 12	4 AABCCCCAAAAA
8 3	6 CCC CCC ACB ACB ABB ABB CCC ABB

Задача C. Non-Squares

Имя входного файла:	`standard input`
Имя выходного файла:	`standard output`
Ограничение по времени:	`2 seconds`
Ограничение по памяти:	512 mebibytes

Given a positive integer n, check whether it can be written as a product of k positive integers that are *not* square numbers.

Формат входного файла

The first line of the input contains an integer t ($1 \le t \le 10$) specifying the number of test cases. Each of the next t lines contains a single test case which is described by two integers n ($1 \le n \le 1\,000\,000\,000$) and k ($2 \le k \le 50$).

Формат выходного файла

Your program should output t lines, one line per test case. The answer for each test case is either the word “`YES`” if $n = a_1 \cdot a_2 \cdots a_k$ for some positive integers a_i that are not square numbers, or the word “`NO`” otherwise.

Примеры

standard input	standard output

4
1 2
6 2
7 2
8 3
	NO
	YES
	NO
	YES

Задача D. Stacks of Coins

Имя входного файла:	`standard input`
Имя выходного файла:	`standard output`
Ограничение по времени:	`2 seconds`
Ограничение по памяти:	512 mebibytes

A number of gold and silver coins, each of the same shape, are arranged into n stacks on the table. The stacks may have different heights.

In a single operation we can swap two coins that belong to different stacks. The operation can only be performed if the two coins are located at the corresponding positions in the stacks, that is, at the same height above the table.

A stack is said to be *unicolor* if it is composed of coins of the same color (all gold or all silver). Calculate how many unicolor stacks can be constructed at maximum by performing an arbitrary number of operations described above.

Формат входного файла

The first line of the input contains one integer n ($1 \le n \le 1\,000\,000$) specifying the number of stacks.

Each of the following n lines contains a string composed of letters “`G`” and “`S`” which describes one stack. Here, “`G`” stands for a gold coin, and “`S`” stands for a silver coin. The coins are listed from the surface of the table to the top of the stack.

The total number of coins on the table does not exceed $1\,000\,000$.

Формат выходного файла

Your program should output a single integer: the maximum number of unicolor stacks that can be obtained using the aforementioned swap operation.

Примеры

standard input	standard output
5 GS SG SSG SGS SSSSG	4
3 GGG GG G	3

Задача E. State Roads

Имя входного файла:	`standard input`
Имя выходного файла:	`standard output`
Ограничение по времени:	`5 seconds`
Ограничение по памяти:	512 mebibytes

The ancient Kingdom of Byteland had a road network designed for carriage transportation. Byteland was divided into several states. However, the division of the towns among the states was not fixed in time. Historians are now trying to recover the division at different points of time.

The only evidence the historians have about the examined period are chronicles telling which roads were classified as state roads at certain moments in history. The historians assume that the towns that were connected by a state road belonged to the same state. They are, however, uncertain if every pair of towns from the same state were connected by a state road and even if every pair of such towns were connected by a sequence of state roads (the connection could have used local roads which are not included in the historical data).

To test their methods, historians need a system which tracks the status of the roads in chronological order and answers the queries of the following form: at a given moment in history, given a set of towns, find out whether it is possible that the set of towns that belong to some single state is exactly the given set.

Формат входного файла

The first line of the input contains two integers n and q ($1 \le n \le 1\,000\,000$, $1 \le q \le 2\,000\,000$), which specify the number of towns in Byteland and the number of events (road status changes according to the chronicles and historians' queries). The towns are numberedT from 1 to n.

Next q lines describe events in chronological order (a query corresponds to the moment in history when all road status changes above the query already took place, and all the following road status changes didn't). Each of these lines has one of the following formats:

- "1 u v" means that towns u and v are now connected by a state road ($1 \le u < v \le n$),
- "2 m" means that the road that became a state road during the m-th type 1 event is not a state road anymore (m is between 1 and the number of events of type 1 so far),
- "3 k u_1 u_2 … u_k" represents a query whether the set of towns $\{u_1, u_2, \ldots, u_k\}$ could have formed a single state at the considered moment of time ($1 \le k \le n$, $1 \le u_1 < u_2 < \ldots < u_k \le n$).

Each road in Byteland is bidirectional. There is at most one road between any pair of different towns.

Initially there are no state roads. Any road may become a state road several times. No two events of type 2 share the same value of m. The sum of values of k over all queries does not exceed $2\,000\,000$.

Формат выходного файла

For each query (event of type 3) in the input, your program should output one line with the word "`YES`" if the given set of towns could have been the full list of towns of some state at the respective moment in history, or the word "`NO`" otherwise.

Примеры

standard input	standard output

4 10
3 3 1 3 4
1 1 2
3 3 1 3 4
1 2 3
3 2 1 3
3 3 1 2 3
1 3 4
1 2 4
2 1
3 3 2 3 4
	YES
	NO
	NO
	YES
	YES

Задача F. Stickers

Имя входного файла:	`standard input`
Имя выходного файла:	`standard output`
Ограничение по времени:	`7 seconds`
Ограничение по памяти:	512 mebibytes

In a nearby shop one can buy stickers. There are n kinds of stickers, namely $s_1, \ldots, s_n$, where each s_i is a non-empty word consisting of exactly d upper-case letters. The shop has infinite supply of each kind of stickers. You have been requested to paste a long word t using the minimum number of stickers, however you are not allowed to cut, or modify in any other sense, any of the stickers.

Formally, one can imagine this process as follows. Initially we have a string z consisting of m characters "`#`" where m is the length of t. In a single operation, we can take any string s_i, choose any integer p between 0 and $m - d$, and for each $j \in \{1, \ldots, d\}$ replace the $(p + j)$-th character of z by the j-th character of s_i (characters in strings are numbered from 1). We want to find the minimum number of such operations needed to produce the word t.

Формат входного файла

The first line of the input contains two integers n and d $(1 \leq n, d \leq 50)$ which specify the number of kinds of stickers and the length of each sticker, respectively.

Each of the following n lines contains a string composed of d letters from $\{\mathtt{A}, \ldots, \mathtt{Z}\}$, representing a single sticker.

The last line of the input contains a single word t consisting of at least 1 and at most 10 000 letters from $\{\mathtt{A}, \ldots, \mathtt{Z}\}$.

Формат выходного файла

Your program should output a single integer: the minimum number of stickers needed to paste the word t. If pasting t is impossible, your program should output the word "`NO`" instead.

Примеры

standard input	standard output
2 3 ABA BCB ABACABA	3
2 4 ABBA BCBA ABBCBBA	NO