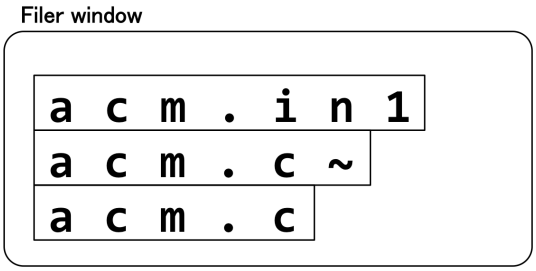
Problem C. Delete Files

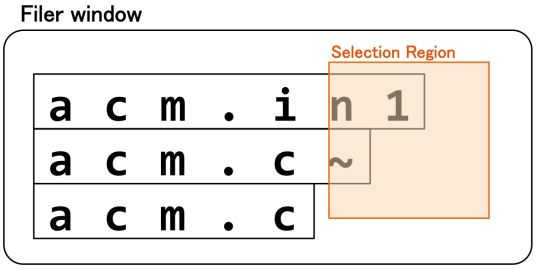
Input file:	*standard input*
Output file:	*standard output*
Time limit:	2 seconds
Memory limit:	512 mebibytes

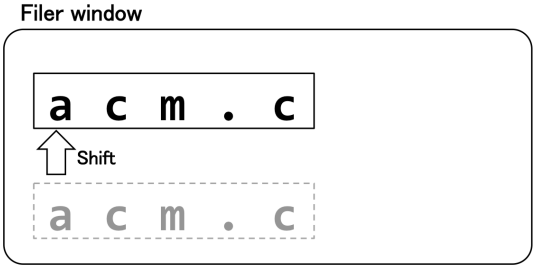
You are using an operating system named "Jaguntu". Jaguntu provides "Filer", a file manager with a graphical user interface.

When you open a folder with Filer, the name list of files in the folder is displayed on a Filer window. Each filename is displayed within a rectangular region, and this region is called a filename region. Each filename region is aligned to the left side of the Filer window. The height of each filename region is 1, and the width of each filename region is the filename length. For example, when three files "`acm.in1`", "`acm.c `", and "`acm.c`" are stored in this order in a folder, it looks like this on the Filer window.



You can delete files by taking the following steps. First, you select a rectangular region with dragging a mouse. This region is called selection region. Next, you press the delete key on your keyboard. A file is deleted if and only if its filename region intersects with the selection region. After the deletion, Filer shifts each filename region to the upside on the Filer window not to leave any top margin on any remaining filename region. For example, if you select a region from the first picture below, then the two files "`acm.in1`" and "`acm.c `" are deleted, and the remaining file "`acm.c`" is displayed on the top of the Filer window as at the second picture.





You are opening a folder storing N files with Filer. Since you have almost run out of disk space, you want to delete unnecessary files in the folder. Your task is to write a program that calculates the minimum number of times to perform deletion operation described above.

Input

The input consists of a single test case.

The first line contains one integer N ($1 \leq N \leq 1,000$), which is the number of files in a folder. Each of the next N lines contains a character D_i and an integer L_i: D_i indicates whether the i-th file should be deleted or not, and L_i ($1 \leq L_i \leq 1,000$) is the filename length of the i-th file. If D_i is '`y`', the i-th file should be deleted. Otherwise, D_i is always '`n`', and you should not delete the i-th file.

Output

Output the minimum number of deletion operations to delete only all the unnecessary files.

Examples

standard input	standard output
3 y 7 y 6 n 5	1
3 y 7 n 6 y 5	2
6 y 4 n 5 y 4 y 6 n 3 y 6	2

Problem C. Jump

Input file:	*standard input*
Output file:	*standard output*
Time limit:	2 seconds
Memory limit:	512 mebibytes

Snuke is standing on an infinitely long road.

The position on this road is represented by a real number.

Snuke can perform N types of jumps. The jump of type i is symmetric with respect to the point a_i. That is, if he performs this jump at point x, he will jump to $2a_i - x$).

You are given Q queries. In the i-th query, you are asked to compute the minimum number of jumps Snuke must perform to go from s_i to t_i. If t_i is unreachable from s_i by performing a series of jumps, print -1 instead.

Input

First line of the input contains one integer N ($1 \le N \le 200$). Next N lines contain integers a_i, one per line ($0 \le a_1 < \ldots < a_N \le 10^4$). Next line contains one integer Q — the number of queries ($0 \le Q \le 10^5$). Each of the next Q lines contains one query and consists of two integers s_i and t_i ($0 \le s_i, t_i \le 10^4$).

Output

For each query, print the answer in a single line.

Example

standard input	standard output
4 1 2 4 7 10 2 3 5 6 6 0 3 7 10 3 7 6 5 5 2 10 4 10 10 10	-1 -1 2 2 -1 -1 0 3 1 0

Problem I. Robots

Input file:	*standard input*
Output file:	*standard output*
Time limit:	2 seconds
Memory limit:	512 mebibytes

Snuke has N robots. They are numbered 1 through N. Initially, the robot i is placed at (x_i, y_i), and the direction the robot is initially facing is d_i. Here, d_i is one of '`U`', '`D`', '`L`', and '`R`': they represent the y-plus direction, the y-minus direction, the x-minus direction, and the x-plus direction, respectively. Robots and Snuke are considered points on the plane.

Initially, no robots are moving. However, when a robot is touched by something (Snuke or another robot), it will immediately start moving in the direction it is facing with unit speed. These robots are made of strange material and they can pass through other robots. Once a robot starts moving, it keeps moving no matter what happens; even if it touches another robot, it won't change its direction and speed.

Snuke touched the robot 1 at time 0. Compute the coordinates of each robot at time T.

Input

First line of input contains two integers N and T ($1 \leq N \leq 10^5$, $0 \leq T \leq 10^{18}$). The i-th of next N lines contains two integers x_i and y_i and letter d_i — initial coordinates and direction of i-th robot ($0 \leq x_i, y_i \leq 10^9$, d_i is one of the following characters: '`U`', '`D`', '`L`', '`R`'). At time 0, no two robots are at the same position.

Output

Print N lines. In the i-th line, print the coordinates of the robot i at time T.

Example

standard input	standard output
5 10 1 0 U 3 1 U 1 2 R 1 1 L 0 1 R	1 10 3 6 9 2 -8 1 8 1

Problem K. Toll Roads

Input file:	*standard input*
Output file:	*standard output*
Time limit:	10 seconds
Memory limit:	256 mebibytes

There are n cities in the country of Flatland, $n-1$ bidirectional roads connect some pairs of Flatland's cities. It is possible to get from any city to any other city of the country using roads. In this problem we will refer to the smallest set of roads needed to travel from a to b as a *simple path*.

The government of Flatland has recently introduced payment tolls on all roads. Using one road connecting two cities costs one ruble. That means that every time when you travel from one city to another one, you have to pay as many rubles as the number of roads you pass during your journey.

Many people got upset because of this decision, especially people who travel long distances. Political opponents of the current government claim that the maximal cost of a simple path in Flatland is very high, namely *cost* rubles. The government has decided to support people and calm down the opposition. They want to reduce the maximal cost of a simple path in the country as much as possible. In other words, they want the number *cost* reported by the opposition to be the lowest possible. The government will allow to remove tolls from at most k roads in order to achieve that. If there are multiple ways to do that, they want to minimize the number of roads that will become free.

Like any other government, Flatland's one is quite inflexible. They additionally require that it should be possible to represent the set of roads which will become free as a simple path between some cities x and y. Your task is to help the government.

Input

The first line contains two integers n and k ($1 \le k < n \le 5000$), the number of cities in Flatland and the maximum number of roads to become toll-free, respectively. Each of the next $n-1$ lines contains two integers u_i and v_i, meaning that there is a road connecting cities u_i and v_i ($0 \le u_i, v_i < n$).

Output

On the first line of the output, print one integer: the minimum *cost* of the most expensive simple path in Flatland that the government can achieve ($0 \le cost < n-1$). On the second line, print the minimum number of roads t that must be made free to achieve that *cost* ($0 \le t \le k$). If $t > 0$, print two space-separated integers on the third line: the numbers of cities x and y such that all roads on the simple path between them must be made free ($0 \le x, y < n$, the simple path between x and y must contain exactly t roads).

Examples

standard input	standard output
8 3 0 2 0 5 2 3 5 1 4 5 5 6 6 7	2 3 2 6
5 2 0 1 0 2 0 3 0 4	2 0