Problem D. Simple game

Input file:	`game.in`
Output file:	`game.out`
Time limit:	1 second
Memory limit:	256 megabytes

NurlashKO was well behaved during last year, for this Ded Moroz gifted him for New Year polygonal chain line with N vertices. i-th vertex of this chain located at the point with coordinates (i, y_i).
Very soon a new game with this geometric figure was invented: the following operations are executed M times:

- Change y coordinate for one of the chain vertexes.
- Draw a horizontal line at the height H and count its intersections with the chain. Note, that all points of horizontal line have y coordinate equal to H.

NurlashKO likes this game and he asks your help to write a program for this game.

Input

First line of input contains two positive integers $N, M (1 \le N, M \le 100\,000)$ — the numbers of vertices and operations in the game, respectively.
Next line contains N positive integers separated by a single space $h_i (1 \le h_i \le 1\,000\,000)$ — h_i is the original height of the i-th vertex.
Then in M lines follows descriptions of the game operations in the following format:

- 1 *pos val* $(1 \le pos \le N, 1 \le val \le 1\,000\,000)$ — vertex number and its new new height, respectively.
- 2 H $(1 \le H \le 1\,000\,000)$ — height of the horizontal line. It is guaranteed that this line will never intersects with the chain at the vertices.

Output

For each query of the second type on a separate line output the number of intersections of horizontal line with the chain. Output answers to queries in the same order as they appear in the input file.

Scoring

This problem consists of 3 subtasks:

1. $1 \le N, M \le 1\,000$. Score 22 points.
2. $1 \le N, M \le 100\,000$. Only query (second type) operations are allowed. Score 27 points.
3. $1 \le N, M \le 100\,000$. Score 51 points.

Each subtask will be scored only if the solution successfully passes all of the previous subtasks.

Example

game.in	game.out
3 3 1 5 1 2 3 1 1 5 2 3	2 1

Problem E. Bomb

Input file:	`bomb.in`
Output file:	`bomb.out`
Time limit:	1 second
Memory limit:	128 megabytes

Earthlings and aliens are fighting for Mars. The battle goes on a rectangular grid of cells of size $N \times M$. Each cell is wholly owned by one of the parties. Earthlings may create a bomb destroying every cell in some rectangular area on the battlefield, and sides of this area are parallel to sides of battlefield. Bomb can not be rotated and used outside of the battlefield. Bomb can be used unlimited number of times. Of course, humans do not want to destroy their own cells, however they can create a bomb with only a certain size. Calculate bomb with maximal affected area (i.e. the product of the height and width), such that it is possible to destroy all cells of aliens and do not destroy any Earthlings cell. Any alien cell can be destroyed multiple times.

Input

First line of input contains two integers N, M ($1 \leq N, M \leq 2\,500$), separated by space, where N and M — height and width of battlefield correspondingly. Then N lines each one by M symbols follows, defining battlefield. If symbol in given line is «0», then corresponding cell belongs to Earthlings, otherwise if symbol is «1», then cell belongs to aliens.

Output

Output one integer - the maximal area of bomb destruction

Scoring

This task contains exactly 100 tests:

1. In tests `1-6`: $N = 1$ or $M = 1$.
2. In tests `7-16`: $1 \leq N, M \leq 20$.
3. In tests `17-26`: $1 \leq N, M \leq 100$.
4. In tests `27-36`: $1 \leq N, M \leq 450$.
5. In tests `37-100`: $1 \leq N, M \leq 2500$.

For each successfully passed test participant get 1 point.

Example

bomb.in	bomb.out
5 6 001000 111110 111110 111110 000100	3

Note

In sample test size of optimal bomb is 3×1.

Problem F. Hard route

Input file:	`road.in`
Output file:	`road.out`
Time limit:	1 second
Memory limit:	256 megabytes

Mansur — is governor of the country ACMstan. There are N cities and $N-1$ two-way roads in this country. It is known that from any city you can go to any other city moving along existing roads. More formally, the country looks like a tree, where the vertices are cities and the edges are two-way roads.
Also, in this country, the cities with **exactly one** connected road called *terminal*. A route is a simple path from one *terminal* to another *terminal*. The distance between two cities is the minimum number of roads on the way between them. The distance from city to the route is the minimum number of roads on the way from given city to any city on the route. Mansur decided to implement **exactly one** route in ACMstan, however he interested in only *hard* routes. Hardness of route computed as follows: let A and B are *terminals* of the route, and H is the maximum distance from any city in the country to this route, then the *hardness* of route is product of H and the distance between A and B.
Mansur asked Temirulan to find maximal *hardness* over all routes, in fact he is interested to know the number of such routes. Temirulan asking help from you.
It's strongly recommend to read explanation below.

Input

First line of input contains a positive integer N ($2 \le N \le 500000$) — the number of cities in the country. The cities are numbered from 1 to N. The following $N-1$ lines contain 2 positive integers, separated by single space, u_i, v_i ($1 \le u_i, v_i \le N; u_i \ne v_i$) — road connecting cities u_i and v_i. It is guaranteed that the given graph is a tree.

Output

Output in single line two integers — the maximal *hardness* and the number of routes, separated by single spaces. **Note that**, route from A to B and route from B to A are **the same** routes.

Scoring

This problem consists of three subtasks:

1. $2 \le N \le 100$. Score 19 points.
2. $2 \le N \le 5000$. Score 33 points.
3. $2 \le N \le 500000$. Score 48 points.

Each subtask will be scored if only if the solution successfully passes all of the previous subtasks.

Examples

road.in	road.out
7 1 2 1 3 2 4 2 5 3 6 3 7	6 2
4 1 2 2 3 2 4	2 3
5 1 2 2 3 3 4 4 5	0 1

Note

A simple path is a path with no repeated vertices. **Note**, that there may be simple path which is not route.

First sample test:

There is four *terminal* cities with number 4, 5, 6 and 7. For route 4-2-1-3-6, the distance is 4 and distances from other cities to this route is [1, 1], maximum among them is 1, so the *hardness* of route is equal to $4 \times 1 = 4$. For route 4-2-5, the distance is 2, maximum distance among other cities is 3 (from 6 or 7), *hardness* of route equal to $3 \times 2 = 6$. *Hardness* of 6-3-7 also 6, but other routes has smaller *hardness*.

In third sample test there is only two *terminal* cities 1 and 5, so there is exactly one route 1-2-3-4-5, the distance is 4 and maximum distance among all cities to route is 0, because all cities are on this route already. *Hardness* is equal to $4 \times 0 = 0$.