A. Picky Professor Party

time limit per test: 1 second
memory limit per test: 256 megabytes

The professors are having a party, and of course, beer will be included. There are n professors and m types of beer. The i^{th} professor would like to drink exactly p_i units of beer. There are exactly b_j units of the j^{th} type of beer. Each professor has strict preferences about the types of beer that they will drink. Help them determine whether there is enough beer to satisfy them all.

Input

The first line of input contains two integers n $(1 \le n \le 100)$, the number of professors, and m $(1 \le m \le 100)$, the number of types of beer. The second line contains n integers, the i^{th} of which is p_i $(1 \le p_i \le 100)$, the number of units of beer that professor i wants to drink. The third line contains m integers, the j^{th} of which is b_j $(1 \le b_j \le 100)$, the number of available units of the j^{th} type of beer. The final n line contains the professors beer preferences. The i^{th} such line begins with an integer k_i $(1 \le k_i \le m)$, denoting the number of types of beer that professor i is willing to drink, followed by k_i integers, which are the indices of those beers. Indices of beers are 1-based.

Output

Display `PARTY` if there is enough beer to satisfy all of the professors. Otherwise, display `NO`.

Examples

input

```
3 2
1 1 2
2 2
1 1
1 2
2 1 2
```

output

```
PARTY
```

input

```
3 2
2 2 2
2 2
1 1
1 2
2 1 2
```

output

```
NO
```

input

```
3 2
2 2 3
2 3
1 1
1 1
1 2
```

output

```
NO
```

B. Devious Dominos

time limit per test: 5 seconds
memory limit per test: 256 megabytes

You're given a rectangular board of square cells. Some of the cells are blocked. You are to compute the maximum number of dominos (1 × 2 tiles) that can be placed on the board. Each domino will be in either a vertical or horizontal orientation, and is placed on two neighboring squares, neither of which is blocked.

Input

The first line contains two space-separated integers: r and c. The next r lines are strings of length c comprised of the characters "." and "x". The "x" characters denote cells of the board that are blocked. $1 \leq r, c \leq 30$.

Output

Print the maximum number of dominos that can be placed on the board satisfying these constraints.

Examples

input

```
2 3
...
...
```

output

```
3
```

input

```
5 4
.xx.
xx.x
x...
xx.x
.xx.
```

output

```
1
```

C. Programmers Perpendicular Processors

time limit per test: 1 second
memory limit per test: 256 megabytes

On an $n \times m$ chessboard, each cell is a either a programmer, or a computer, or empty. Assign computers to programmers, such that

1. Each computer is assigned to at most one programmer;
2. If a computer is assigned to a programmer, they must share a common edge on the chessboard.

A programmer is happy if there are two computers assigned to him/her, and these two computers are neither on the same row nor on the same column. Maximize the number of happy programmers.

Input

The first line contains two integers n and m. Each of the following n lines contains m integers, which could be 0, 1 or 2. Here 0 means the corresponding cell is empty, 1 means a programmer and 2 means a computer.

Output

The maximum number of happy programmers.

Example

input	output
3 4 2 1 2 2 2 2 1 0 1 2 2 2	2

D. Costflow Competition

time limit per test: 1 second
memory limit per test: 256 megabytes

Given a directed flow network, suppose each edge (arc) has a positive weight. The cost of a flow f is defined to be $\sum_{e \in E} f(e)w(e)$, where $f(e)$ is the flow on edge e and $w(e)$ is the positive weight of edge e.

For a given directed flow network with n vertices and m edges (arcs), together with a positive integer P, Alice first chooses a maximum flow f, then according to Alice's choice of f, Bob assigns a weight to each edge under the constraint that $w(e) \geq 0$ for all edges and $\sum_{e \in E} w(e) = P$.

Alice's goal is to minimize the cost of f, while Bob's goal is to maximize to cost. Suppose both players act optimally, output the cost of the maximum flow chosen by Alice.

Input

The first line contains three integer n, m and P. Each of the following line contains three integers u, v and c, which denotes a directed edge from the u-th vertex to the v-th vertex with capacity c. It is guaranteed that $1 \leq u, v, \leq n$ and $1 \leq c \leq 50000$.

It is also guaranteed that $1 \leq n \leq 100$, $1 \leq m \leq 1000$ and $1 \leq P \leq 10$ and there is no self-loop in the flow network.

Output

Output the cost of the maximum flow. Any solution within 10^{-4} of the correct answer will be considered correct.

Example

input	output
3 2 1 1 2 10 2 3 15	9.9999999999

E. Making My Tree Minimum

time limit per test: 1 second

memory limit per test: 256 megabytes

Given a weighted connected undirected graph $G = (V, E, w)$ with n vertices and m edges, together with a spanning tree T of G. Find a new set of weights w', such that T is a minimum spanning tree of G with respect to w', while minimizing the cost which is defined to be $\sum_{e \in E} |w(e) - w'(e)|$.

Input

The first line contains two integer n and m. Each of the following m lines contains three integers u, v and w, which means there is an edge (u, v) with weight w in G. It is guaranteed that $1 \leq u, v \leq n$ and $0 \leq w \leq 10000$. Each of the following $n - 1$ lines contains two integers u and v, which means there is an edge (u, v) in the spanning tree T. It is guaranteed that (u, v) also belongs to G.

It is guaranteed that G is a simple graph and $n \leq 50$

Output

The minimum cost. Is is guaranteed that the answer is an integer.

Example

input

```
6 9
1 2 2
1 3 2
2 3 3
3 4 3
1 5 1
2 6 3
4 5 4
4 6 7
5 6 6
1 3
2 3
3 4
4 5
4 6
```

output

```
8
```