Network Flow Applications

A. Network Flow (Easy)

time limit per test: 2 seconds
memory limit per test: 1024 megabytes
input: standard input
output: standard output

Find the maximum flow through a graph from the source to the sink. You may not assume anything about the graph (there may be loops, there may be multiple edges between two vertices, etc)

(Any max flow algorithm should pass this problem)

Input

The first line will contain two integers, n ($2 \le n \le 100$) and m ($1 \le m \le 1\,000$), denoting the number of nodes and the number of edges.

The second line will contain two integers s and t ($1 \le s, t \le n$) denoting the source and sink ($s \neq t$).

The next m lines contain 3 integers u, v ($1 \le u, v \le n$) and c ($0 \le c \le 10\,000$) denoting an edge in the graph.

Output

Output the maximum flow of the graph from s to t.

Examples

input
2 1 1 2 1 2 3
output
3

input
4 5 1 4 1 2 10 1 3 10 2 3 1 2 4 10 3 4 10
output
20

B. Soldier and Traveling

time limit per test: 1 second
memory limit per test: 256 megabytes
input: standard input
output: standard output

In the country there are n cities and m bidirectional roads between them. Each city has an army. Army of the i-th city consists of a_i soldiers. Now soldiers roam. After roaming each soldier has to either stay in his city or to go to the one of neighboring cities by at **moving along at most one road**.

Check if is it possible that after roaming there will be exactly b_i soldiers in the i-th city.

Input

First line of input consists of two integers n and m ($1 \le n \le 100$, $0 \le m \le 200$).

Next line contains n integers $a_1, a_2, ..., a_n$ ($0 \le a_i \le 100$).

Next line contains n integers $b_1, b_2, ..., b_n$ ($0 \le b_i \le 100$).

Then m lines follow, each of them consists of two integers p and q ($1 \le p, q \le n, p \ne q$) denoting that there is an undirected road between cities p and q.

It is guaranteed that there is at most one road between each pair of cities.

Output

If the conditions can not be met output single word "NO".

Otherwise output word "YES" and then n lines, each of them consisting of n integers. Number in the i-th line in the j-th column should denote how many soldiers should road from city i to city j (if $i \ne j$) or how many soldiers should stay in city i (if $i = j$).

If there are several possible answers you may output any of them.

Examples

input
4 4 1 2 6 3 3 5 3 1 1 2 2 3 3 4 4 2

output
YES 1 0 0 0 2 0 0 0 0 5 1 0 0 0 2 1

input
2 0 1 2 2 1

output
NO

D. Array and Operations

time limit per test: 1 second
memory limit per test: 256 megabytes
input: standard input
output: standard output

You have written on a piece of paper an array of n positive integers $a[1], a[2], ..., a[n]$ and m *good* pairs of integers $(i_1, j_1), (i_2, j_2), ..., (i_m, j_m)$. Each *good* pair (i_k, j_k) meets the following conditions: $i_k + j_k$ is an odd number and $1 \le i_k < j_k \le n$.

In one operation you can perform a sequence of actions:

- take one of the *good* pairs (i_k, j_k) and some integer v ($v > 1$), which divides both numbers $a[i_k]$ and $a[j_k]$;
- divide both numbers by v, i. e. perform the assignments: $a[i_k] = \frac{a[i_k]}{v}$ and $a[j_k] = \frac{a[j_k]}{v}$.

Determine the maximum number of operations you can sequentially perform on the given array. Note that one pair may be used several times in the described operations.

Input

The first line contains two space-separated integers n, m ($2 \le n \le 100$, $1 \le m \le 100$).

The second line contains n space-separated integers $a[1], a[2], ..., a[n]$ ($1 \le a[i] \le 10^9$) — the description of the array.

The following m lines contain the description of *good* pairs. The k-th line contains two space-separated integers i_k, j_k ($1 \le i_k < j_k \le n$, $i_k + j_k$ is an odd number).

It is guaranteed that all the *good* pairs are distinct.

Output

Output the answer for the problem.

Examples

input
3 2 8 3 8 1 2 2 3
output
0

input
3 2 8 12 8 1 2 2 3
output
2

E. Network Flow

time limit per test: 2 seconds
memory limit per test: 1024 megabytes
input: standard input
output: standard output

Find the maximum flow through a graph from the source to the sink. You may not assume anything about the graph.

Input

The first line will contain two integers, n ($2 \le n \le 10\ 000$) and m ($1 \le m \le 200\ 000$), denoting the number of nodes and the number of edges.

The second line will contain two integers s and t ($1 \le s, t \le n$) denoting the source and sink ($s \neq t$).

The next m lines contain 3 integers u, v ($1 \le u, v \le n$) and c ($0 \le c \le 10^9$) denoting an edge in the graph.

Output

Output the maximum flow of the graph from s to t.

Examples

input
2 1 1 2 1 2 3
output
3

input
4 5 1 4 1 2 10 1 3 10 2 3 1 2 4 10 3 4 10
output
20

Note

Use this code for your input:

```
int n,m,s,t;
scanf("%d%d%d%d",&n,&m,&s,&t);
for(int i=0;i<m;i++){
  int u,v,c;
  scanf("%d%d%d",&u,&v,&c);
  add_edge(u,v,c); // You may want u-1,v-1...
}
```