2011 Canadian Computing Competition
Day 1, Problem 1
Putnam

Input: from standard input
Output: to standard output
Source file: `putnam.{c,cc,pas}`

Problem Description
After your great performance in computing competitions, you have been admitted to the University of Waterloo. In your first term there, full of enthusiasm, and wanting to increase your streak of success in academic competitions, you decide to take part in the Putnam competitiion.

Three months later, you get to know your score. It is very good, so you decide to add it to your CV. However, the released results only include the average rank for each of the attained scores. You think that might not be enough information, and want to include the exact range in which your score falls (i.e. if there are 25 people with scores better than yours, and 3 people tied with you, the range would be 26-29).

Furthermore, you want to build a program that would handle possible future situations in which the range of the scores, as well as the number of contestants, have greatly increased. You also want your program to work even if the scores are not necessarily given to you in order.

Input Specification
The first line of the input file will contain an integer N between 1 and 100000, inclusive.

N lines will follow, containing the scores that were attained in the contest, as well as the average rank for each of them. Each of them will contain two numbers, separated by a space. The first number will be between 0 and $3 * 10^9$, inclusive. It will correspond to a score that was attained in the contest by some people. The second number will be a decimal number between 0 and $3 * 10^8$ specified using the format in the sample input, and it will contain the average rank corresponding to that score.

The last line of the input contains your score in the competition. You may assume that this number appears as the first number in one of the previous N lines.

Output Specification
You will output two lines. They will contain the range corresponding to the rank of your score.

Sample Input
```
6
5 2
```

```
4 10
3 20.5
1 34
0 35
2 29
4
```

Output for Sample Input

```
4
16
```

2011 Canadian Computing Competition
Day 1, Problem 2
Vampire Tunnels

Input: from standard input
Output: to standard output
Source file: `vampire.{c,cc,pas}`

Problem Description

You are a vampire, and you want to travel from some point A to another point B. You may travel in the sunshine above ground or avoid the sunshine by travelling underground via certain tunnels. You have mapped out the area you wish to travel, and have found some secret tunnels between some points, some other points that you can walk between above ground. Both the tunnels and above ground paths are bidirectional. Given that you can't be exposed to the sunlight for more than S seconds in total ($0 \leq S \leq 3600$), you want to minimize the total travel time (given that you have a constant velocity of 1 unit of distance per second).

Input Specification

On the first line of input, you have the number S, the maximum number of seconds that you can be exposed to the sun. On the next line is the number N ($2 \leq N \leq 1,600$), which is the number of points, and the number E ($1 \leq E \leq 10,000$), which is the number of connections between these N points, separated by one space.

The next E lines each contain information about the connections between points. Specifically, each of these lines contains four integers: s (one end point of a connection) ($0 \leq s \leq N-1$), t (the other end point of a connection) ($0 \leq t \leq N-1$, $s \neq t$), d (the distance between s and t, $1 \leq d \leq 10,000$), u (indicate whether this is underground or above ground: 1 indicates it is above ground, and 0 indicates there is a tunnel between s and t).

Note: for 30% of the marks for this question, you can assume $N \leq 50$.

Output Specification

The output is one integer, which is the minimum amount of time required to get from point 0 to point $N-1$, with the constraint that there are not more than S seconds of exposure to the sun. If there is no possible path which satisfies the constraint, output -1.

Sample Input

```
3
4 6
0 1 3 1
0 2 4 1
```

```
0 3 10 1
1 2 3 0
1 3 1 1
2 3 3 0
```

Output for Sample Input

```
9
```

Explanation of Sample Input

The path $0 \to 1 \to 2 \to 3$ gives a total travel time of 9 seconds with 3 seconds of exposure to the sun. All other paths either required more time (e.g., $0 \to 3$ uses 10 seconds) or overexpose to the sun (e.g., $0 \to 1 \to 3$ exposes to the sun for 4 seconds).

2011 Canadian Computing Competition
Day 1, Problem 3
Spies Like Us

Input: from standard input
Output: to standard output
Source file: `spies.{c,cc,pas}`

Problem Description

The Ultrasecret Spy Organization is very concerned about recent leads concerning a secret conspiracy involving the use of the Comic Sans font.

In order to avoid groupthink, the Ultrasecret Spy Organization has decided to divide its agents in two groups. Each of the two groups will carry its own investigation. However, occassionally interaction between members of different groups will happen through some previously designated contact points (i.e. two people on different teams that are allowed to speak with each other in special circumstances). This has to be made in a way that preserves the fact that there is not much communication between the teams. To make this rule more exact, two people on the same team can have no more than one common contact on the other team.

You are given a plan for the contact points between the two groups. Your task is to determine whether this actually satisfies the constraint that two people on the same team can have no more than one common contact in the other team.

Input Specification

The first line of the input file will contain two space-separated integers N and M, $(1 \leq N, M \leq 2000)$. They represent the number of people in each of the teams. The next line will contain an integer K, $(0 \leq K \leq NM)$. Each of the following K lines will containg two integers i and j, with $(1 \leq i \leq N, 1 \leq j \leq M)$. This input will represent that person i of the first team and person j of the second team are allowed to communicate with each other.

Note: For test cases worth 25% of the marks, you may assume $N, M \leq 200$.

Output Specification

For each input, you will output one line. Its content will be `YES`, if the proposal satifies the constraint that two people on the same team can have no more than one common contact on the other team, and `NO` otherwise.

Sample Input

```
5 3
4
1 1
```

```
2 1
3 1
4 1
```

Output for Sample Input

```
YES
```

2011 Canadian Computing Competition
Day 2, Problem 1
Reorganization

Input: from standard input
Output: to standard output
Source file: `reorg.{c,cc,pas}`

Problem Description

Alice and Bob own a huge company. This company was losing money consistently over the last 30 years, since its owners spent too much time playing games with mathematicians. Alice and Bob decide to make a change.

Alice and Bob start by giving unique employee IDs to each of the n employees ($1 \leq n \leq 100,000$), where each ID I is in the range ($1 \leq I \leq 100,000$).

Then, Alice and Bob give unique ranks to each employee. Each rank R is an integer such that $1 \leq R \leq 10,000,000$. After this, they plan to reorganize the company, by making sure that the employees satisfy the following conditions:

1. There is exactly one director, who has no supervisor;
2. Except for the director, each employee has a supervisor, and this supervisor has a smaller employee ID and a higher rank (the value of rank is smaller); and
3. Each employee can supervise at most 2 people.

Alice and Bob are eager to know whether their company can be reorganized successfully.

Input Specification

The input is a total of $n + 1$ lines. The first line contains n ($1 \leq n \leq 100,000$), indicating the number of employees. On the next n lines are n distinct integers R ($1 \leq R \leq 10,000,000$), one integer per line, where the ith integer indicates the rank of the employee with ID i.

Output Specification

Output `YES` if the company can be reorganized successfully; output `NO` otherwise.

Sample Input 1

```
6
1
6
5
```

```
2
3
4
```

Output for Sample Input 1

```
NO
```

Explanation of Output for Sample Input 1

Employee with rank 1 has employee ID 1, and thus, must be the supervisor. Employees 2 and 3 (with ranks 6 and 5) can only be supervised by employee 1 (with rank 1). However, no other employee (4, 5 or 6) can be supervised by employee 2 or employee 3, since ranks of supervisors must be smaller than the employees they supervise.

Sample Input 2

```
6
1
6
2
3
4
5
```

Output for Sample Input 2

```
YES
```

Explanation for Output for Sample Input 2

Employee 1 (rank 1) supervises both employee 2 (rank 6) and employee 3 (rank 2).

Employee 3 (rank 2) supervises employee 4 (rank 3) and employee 5 (rank 4).

Employee 4 (rank 3) supervises employee 6 (rank 5).

2011 Canadian Computing Competition
Day 2, Problem 2
Fixing Disks

Input: from standard input
Output: to standard output
Source file: `disk.{c,cc,pas}`

Problem Description

You are playing a game with a stack of N disks ($1 \le N \le 100$). The goal of the game is remove all of the disks from your stack. However, there is a cost associated with removing disks, and you wish to minimize the cost of removing all the disks from your stack.

Each disk contains a label, with the label L being in the range $1 \le L \le 20$.

You are also given a "Master stack" of N disks which you can use to help you remove disks from your stack.

You can remove disks from the top of your stack of disks in two ways:

1. remove a disk at the top of your stack: if label of the disk on top of your stack is c, that disk can be removed from your stack with cost c;
2. remove a disk from both the top of the Master stack and the top of your stack if the label is the same between both disks: in this case, there is no cost with removing both disks.

You are also allowed to modify the order of the top K ($1 \le K \le 4$) elements of your stack, so long as immediately after each modification, you remove the top element of your stack. There are three allowed modifications:

1. Reverse: you may reverse the order of the top r ($2 \le r \le K$) disks on your stack. In other words, if the top r disks are $d_1, d_2, ..., d_r$ (reading from the top down), then after this operation, the top r disks will be $d_r, ..., d_2, d_1$ (reading from the top down). The cost of one reverse operation is R ($1 \le R \le 1,000,000$).
2. (Cyclic Shift) Up: you can shift up one disk in the range of the top u disks ($2 \le u \le K$). For example, if the top four disks are d_1, d_2, d_3, d_4 (read from the top down), you can perform an up shift in the range of the top three elements to get d_2, d_3, d_1, d_4 or an up shift of the top four elements to get d_2, d_3, d_4, d_1. The cost of one up shift operation is U ($1 \le U \le 1,000,000$).
3. (Cyclic Shift) Down: you can shift down one disk in the range of the top d disks ($2 \le d \le K$). For example, if the top four disks are d_1, d_2, d_3, d_4 (read from the top down), you can

perform a down shift in the range of the top three elements to get d_3, d_1, d_2, d_4 or a down shift of the top four elements to get d_4, d_1, d_2, d_3. The cost of one down shift operation is D ($1 \leq D \leq 1,000,000$).

If the operations yield a match between the top of the master stack and the top of your stack, you can pop for free. If not, you must pay the cost of the pop.

There is one additional constraint. At any point in the game, if a disk at level j is being popped (levels start at 0 at the bottom of the stack), then all elements that were *originally* at level $j + M$ or higher must have already been popped, where $1 \leq M \leq 5$.

Minimize the cost required to pop all the elements off of your stack.

Input Specification
The first line will contain 6 space-separated integers: N, K, M, D, U, R where:

- N is the number of disks in each of the stacks ($1 \leq N \leq 100$)
- K, the maximum depth at which operations can be done ($1 \leq K \leq 4$)
- M, the threshold before which elements can be removed from our stack, ($1 \leq M \leq 5$)
- D, the cost for a shift in which the bottom element of the selected range goes to the top ($1 \leq D \leq 10^6$)
- U, the cost for an up shift operation (in which the top element of the selected range goes to the bottom), ($1 \leq U \leq 10^6$)
- R, the cost for reversing the top elements of the selected range ($1 \leq R \leq 10^6$)

$2N$ lines will follow, each containing a number L ($1 \leq L \leq 20$). The first N lines will contain the labels of the disks in the Master stack, from top to bottom. The next N lines will contain the labels of the disks in your stack, from top to bottom

Note: for 20% of the marks for this question, you may assume that $K \leq 2$.

Output Specification
Output the integer (on one line) which is the minimal cost required to remove all disks from your stack.

Sample Input
```
7 3 3 4 4 3
5
6
3
```

```
5
4
1
2
3
5
6
5
1
4
1
```

Output for Sample Input

```
5
```

Explanation for Output for Sample Input

We take 3 to the bottom and shift the two blocks below it up, with cost 4. Then we remove four blocks from each stack, remove a 1 from the playing stack, with cost 1, and remove two blocks from each stack.

2011 Canadian Computing Competition
Day 2, Problem 3
Biggest (Zero Carbon) Footprint

Input: from standard input
Output: to standard output
Source file: `bigfoot.{c,cc,pas}`

Problem Description
Having just recently won the lottery, you decide to build a summer resort nestled deep in a forest. However, being a very eco-friendly person, you decide not to cut down any of the trees that grow in the forest. Given a map of the forest and the positions of its trees, determine the area of the largest rectangular plot you can buy that does not contain any of the trees. (Note that your plot must have edges which are parallel to the x and y axes.)

Input Specification
The first line contains n, m, and t ($0 < n, m < 10,000$, $0 < t < 10,000$) representing the dimensions of the given map of the forest and the number of trees indicated on the map respectively. The next t lines each contain two integers x and y ($0 \leq x \leq n, 0 \leq y \leq m$) describing the location of each tree (where $(0, 0)$ is the bottom leftmost point on the map and (n, m) is the top rightmost point on the map).

Note: for 20% of the marks for this question, you may assume that $t \leq 100$, and for 45% of the marks for this question, $t \leq 400$.

Output Specification
Output the area of the largest rectangle that does not contain any of the given trees.

Sample Input
```
5 5 2
1 1
3 3
```

Output for Sample Input
```
12
```