Codeforces Round #299 (Div. 1)

A. Tavas and Karafs

time limit per test: 2 seconds
memory limit per test: 256 megabytes
input: standard input
output: standard output

Karafs is some kind of vegetable in shape of an $1 \times h$ rectangle. Tavaspolis people love Karafs and they use Karafs in almost any kind of food. Tavas, himself, is crazy about Karafs.

Each Karafs has a positive integer height. Tavas has an infinite **1-based** sequence of Karafses. The height of the i-th Karafs is $s_i = A + (i - 1) \times B$.

For a given m, let's define an m-bite operation as decreasing the height of at most m distinct not eaten Karafses by 1. Karafs is considered as eaten when its height becomes zero.

Now SaDDas asks you n queries. In each query he gives you numbers l, t and m and you should find the largest number r such that $l \leq r$ and sequence $s_l, s_{l+1}, \ldots, s_r$ can be eaten **by performing m-bite no more than t times** or print -1 if there is no such number r.

Input

The first line of input contains three integers A, B and n ($1 \leq A, B \leq 10^6$, $1 \leq n \leq 10^5$).

Next n lines contain information about queries. i-th line contains integers l, t, m ($1 \leq l, t, m \leq 10^6$) for i-th query.

Output

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

Sample test(s)

input
2 1 4 1 5 3 3 3 10 7 10 2 6 4 8
output
4 -1 8 -1

input
1 5 2 1 5 10 2 7 4
output
1 2

B. Tavas and Malekas

time limit per test: 2 seconds
memory limit per test: 256 megabytes
input: standard input
output: standard output

Tavas is a strange creature. Usually "zzz" comes out of people's mouth while sleeping, but string s of length n comes out from Tavas' mouth instead.

Today Tavas fell asleep in Malekas' place. While he was sleeping, Malekas did a little process on s. Malekas has a favorite string p. He determined all positions $x_1 < x_2 < ... < x_k$ where p matches s. More formally, for each x_i $(1 \le i \le k)$ he condition $s_{x_i}s_{x_i+1}... s_{x_i+|p|-1} = p$ is fullfilled.

Then Malekas wrote down one of subsequences of $x_1, x_2, ... x_k$ (possibly, he didn't write anything) on a piece of paper. Here a sequence b is a subsequence of sequence a if and only if we can turn a into b by removing some of its elements (maybe no one of them or all).

After Tavas woke up, Malekas told him everything. He couldn't remember string s, but he knew that both p and s only contains lowercase English letters and also he had the subsequence he had written on that piece of paper.

Tavas wonders, what is the number of possible values of s? He asked SaDDas, but he wasn't smart enough to solve this. So, Tavas asked you to calculate this number for him.

Answer can be very large, so Tavas wants you to print the answer modulo $10^9 + 7$.

Input

The first line contains two integers n and m, the length of s and the length of the subsequence Malekas wrote down ($1 \le n \le 10^6$ and $0 \le m \le n - |p| + 1$).

The second line contains string p ($1 \le |p| \le n$).

The next line contains m space separated integers $y_1, y_2, ..., y_m$, Malekas' subsequence ($1 \le y_1 < y_2 < ... < y_m \le n - |p| + 1$).

Output

In a single line print the answer modulo $1000\,000\,007$.

Sample test(s)

input
6 2 ioi 1 3

output
26

input
5 2 ioi 1 2

output
0

Note

In the first sample test all strings of form "ioioi?" where the question mark replaces arbitrary English letter satisfy.

Here $|x|$ denotes the length of string x.

Please note that it's possible that there is no such string (answer is 0).

C. Tavas and Pashmaks

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

Tavas is a cheerleader in the new sports competition named "Pashmaks".

This competition consists of two part: swimming and then running. People will immediately start running R meters after they finished swimming exactly S meters. A winner is a such person that nobody else finishes running before him/her (there may be more than one winner).

Before the match starts, Tavas knows that there are n competitors registered for the match. Also, he knows that i-th person's swimming speed is s_i meters per second and his/her running speed is r_i meters per second. Unfortunately, he doesn't know the values of R and S, but he knows that they are real numbers greater than 0.

As a cheerleader, Tavas wants to know who to cheer up. So, he wants to know all people that might win. We consider a competitor might win if and only if there are some values of R and S such that with these values, (s)he will be a winner.

Tavas isn't really familiar with programming, so he asked you to help him.

Input

The first line of input contains a single integer n ($1 \le n \le 2 \times 10^5$).

The next n lines contain the details of competitors. i-th line contains two integers s_i and r_i ($1 \le s_i, r_i \le 10^4$).

Output

In the first and the only line of output, print a sequence of numbers of possible winners in increasing order.

Sample test(s)

input
3 1 3 2 2 3 1
output
1 2 3

input
3 1 2 1 1 2 1
output
1 3

D. Tavas in Kansas

time limit per test: 2 seconds
memory limit per test: 512 megabytes
input: standard input
output: standard output

Tavas lives in Kansas. Kansas has n cities numbered from 1 to n connected with m bidirectional roads. We can travel from any city to any other city via these roads. Kansas is as strange as Tavas. So there may be a road between a city and itself or more than one road between two cities.

Tavas invented a game and called it "Dashti". He wants to play Dashti with his girlfriends, Nafas.

In this game, they assign an arbitrary integer value to each city of Kansas. The value of i-th city equals to p_i.

During the game, Tavas is in city s and Nafas is in city t. They play in turn and Tavas goes first. A player in his/her turn, must choose a non-negative integer x and his/her score increases by the sum of values of all cities with (shortest) distance no more than x from his/her city. Each city may be used once, or in the other words, after first time a player gets score from a city, city score becomes zero.

There is an additional rule: the player must choose x such that he/she gets the point of at least one city that was not used before. Note that city may initially have value 0, such city isn't considered as been used at the beginning of the game, i. e. each player may use it to fullfill this rule.

The game ends when nobody can make a move.

A player's score is the sum of the points he/she earned during the game. The winner is the player with greater score, or there is a draw if players score the same value. Both players start game with zero points.

If Tavas wins, he'll break his girlfriend's heart, and if Nafas wins, Tavas will cry. But if their scores are equal, they'll be happy and Tavas will give Nafas flowers.

They're not too emotional after all, so they'll play optimally. Your task is to tell Tavas what's going to happen after the game ends.

Input

The first line of input contains two integers n and m ($2 \le n \le 2000$, $n - 1 \le m \le 10^5$).

The second line of input contains two integers s and t ($1 \le s, t \le n$, $s \ne t$).

The next line contains n integers $p_1, p_2, ..., p_n$ separated by spaces ($|p_i| \le 10^9$).

The next m lines contain the roads. Each line contains three integers v, u, w and it means that there's an road with length w between cities v and u ($1 \le u, v \le n$ and $0 \le w \le 10^9$). The road may lead from the city to itself, there may be several roads between each pair of cities.

Output

If Tavas wins, print "`Break a heart`". If Nafas wins print "`Cry`" and if nobody wins (i. e. the game ended with draw) print "`Flowers`".

Sample test(s)

input

```
4 4
1 2
3 2 5 -11
1 4 2
3 4 2
3 1 5
3 2 1
```

output

```
Cry
```

input

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

output

```
Break a heart
```

input

```
2 1
1 2
-5 -5
1 2 10
```

output

```
Flowers
```

E. Tavas on the Path

time limit per test: 3 seconds
memory limit per test: 256 megabytes
input: standard input
output: standard output

Tavas lives in Tavaspolis. Tavaspolis has n cities numbered from 1 to n connected by $n - 1$ bidirectional roads. There exists a path between any two cities. Also each road has a length.

Tavas' favorite strings are binary strings (they contain only 0 and 1). For any binary string like $s = s_1 s_2 \ldots s_k$, $T(s)$ is its $Goodness$. $T(s)$ can be calculated as follows:

Consider there are exactly m blocks of 1s in this string (a block of 1s in s is a maximal consecutive substring of s that only contains 1) with lengths $x_1, x_2, \ldots, x_m$.

Define $T(s) = \sum_{i=1}^{m} f_{x_i}$ where f is a given sequence (if $m = 0$, then $T(s) = 0$).

Tavas loves queries. He asks you to answer q queries. In each query he gives you numbers v, u, l and you should print following number:

Consider the roads on the path from city v to city u: $e_1, e_2, \ldots, e_x$.

Build the binary string b of length x such that: $b_i = 1$ if and only if $l \le w(e_i)$ where $w(e)$ is the length of road e.

You should print $T(b)$ for this query.

Input

The first line of input contains integers n and q ($2 \le n \le 10^5$ and $1 \le q \le 10^5$).

The next line contains $n - 1$ space separated integers $f_1, f_2, \ldots, f_{n-1}$ ($|f_i| \le 1000$).

The next $n - 1$ lines contain the details of the roads. Each line contains integers v, u and w and it means that there's a road between cities v and u of length w ($1 \le v, u \le n$ and $1 \le w \le 10^9$).

The next q lines contain the details of the queries. Each line contains integers v, u, l ($1 \le v, u \le n$, $v \ne u$ and $1 \le l \le 10^9$).

Output

Print the answer of each query in a single line.

Sample test(s)

input

```
2 3
10
1 2 3
1 2 2
1 2 3
1 2 4
```

output

```
10
10
0
```

input

```
6 6
-5 0 0 2 10
1 2 1
2 3 2
3 4 5
4 5 1
5 6 5
1 6 1
1 6 2
1 6 5
3 6 5
4 6 4
1 4 2
```

output

```
10
-5
```

```
-10
-10
-5
0
```
