Problem A
Olympic Ranking

Time limit: 1 second
Memory limit: 1024 megabytes

Problem Description

The Olympic Games are the most important sporting events in human history. Usually, the Olympic Games are held every four years. However, the 2020 Olympic Games just finished in August 2021 due to the COVID-19 outbreaks.

There are more than 200 nations participating the Olympic Games. Thousands of athletes around the world compete in various sports. The atheletes represent their countries or National Olympuc Committees (NOCs) to compete for medals. Therefore, each country or NOC may win medals in the competitions.

There are three types of medals: gold medals, silver medals, and bronze medals. Typically, the gold medals are awarded to the winners of the competitions, and silver medals are awarded to the runner-ups. Most of the bronze medals are awarded to the second runner-ups. However, there can be no second runner-up in some sports. The bronze medals can be awarded in different manners. For example, in a few tournament sports, such as wrestling, boxing, and judo, two bronze medals are awarded to the eliminated semi-finalists.

A country or NOC has a better rank than another country or NOC if one of the following conditions holds.

1. It wins more gold medals.
2. It wins the same ammount of gold medals, and it wins more silver medals.
3. It wins the same ammount of gold medals and silver medals, and it wins more bronze medals.

Please write a program to find the country or NOC which has the best rank among all countries and NOCs.

Input Format

The first line of the input contains one positive integer n. Then n lines follow. Each of the following lines contains three non-negative integers g, s, b, and the name of a country or NOC. They are separated by blanks.

Output Format

Print the name of the country or NOC of the best rank.

Technical Specification

- $1 \leq n < 300$

- $g, s, b \in \{0, 1, \ldots, 999\}$
- There is only one country or NOC of the best rank.
- The names of countries and NOCs consists of only printable ASCII characters.
- The size of an input file does not exceed 3 megabytes.

Sample Input 1

```
4
22 21 22 Great Britain
27 14 17 Japan
39 41 33 United States of America
20 28 23 ROC
```

Sample Output 1

```
United States of America
```

Sample Input 2

```
3
999 999 998 Malaysia
999 999 999 Thailand
999 998 999 Indonesia
```

Sample Output 2

```
Thailand
```

Problem B
Aliquot Sum

Time limit: 10 seconds
Memory limit: 1024 megabytes

Problem Description

A divisor of a positive integer n is an integer d where $m = \frac{n}{d}$ is an integer. In this problem, we define the aliquot sum $s(n)$ of a positive integer n as the sum of all divisors of n other than n itself. For examples, $s(12) = 1 + 2 + 3 + 4 + 6 = 16$, $s(21) = 1 + 3 + 7 = 11$, and $s(28) = 1 + 2 + 4 + 7 + 14 = 28$.

With the aliquot sum, we can classify positive integers into three types: abundant numbers, deficient numbers, and perfect numbers. The rules are as follows.

1. A positive integer x is an abundant number if $s(x) > x$.
2. A positive intewer y is a deficient number if $s(y) < y$.
3. A positive integer z is a perfect number if $s(z) = z$.

You are given a list of positive integers. Please write a program to classify them.

Input Format

The first line of the input contains one positive integer T indicating the number of test cases. The second line of the input contains T space-separated positive integers $n_1, \ldots, n_T$.

Output Format

Output T lines. If n_i is an abundant number, then print `abundant` on the i-th line. If n_i is a deficient number, then print `deficient` on the i-th line. If n_i is a perfect number, then print `perfect` on the i-th line.

Technical Specification

- $1 \leq T \leq 10^6$
- $1 \leq n_i \leq 10^6$ for $i \in \{1, 2, \ldots, T\}$.

Sample Input 1

```
3
12 21 28
```

Sample Output 1

```
abundant
deficient
perfect
```

Almost blank page

Problem C
A Sorting Problem

Time limit: 3 seconds

Memory limit: 1024 megabytes

Problem Description

You are given an array $[p[1],\ p[2],\ ...,\ p[n]]$ where all the numbers in the array are distinct. In addition, the numbers are positive integers between 1 and n. You can only perform the following operations on the array: Pick two indices x and y such that $|p[x]-p[y]| = 1$, and then swap the values of $p[x]$ and $p[y]$. We now want to sort this array in ascending order. That is, to make $p[i] = i$ for all $i \in \{1, 2, \ldots, n\}$. For example, we can sort the array $[p[1] = 2, p[2] = 3, p[3] = 1]$ in two operations:

1. Swap $p[1]$ and $p[3]$. The array becomes $[p[1] = 1, p[2] = 3, p[3] = 2]$.
2. Swap $p[2]$ and $p[3]$. The array becomes $[p[1] = 1, p[2] = 2, p[3] = 3]$ which is sorted in ascending order.

Please write a program to compute the minimum number of operations to sort a given array in ascending order.

Input Format

The input contain two lines. The first line contains one integer n. The second lines contain n space-saparated numbers $p[1], p[2], \ldots, p[n]$ representing the array $[p[1],\ p[2], \ldots,\ p[n]]$

Output Format

Output only one number that denotes the minimum number of operations required to sort the given array.

Technical Specification

- $1 < n \leq 200000$.
- $1 \leq p[i] \leq n$.
- All $p[i]$ are distinct.

Sample Input 1

```
3
1 3 2
```

Sample Output 1

```
1
```

Sample Input 2

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

Sample Output 2

```
7
```

Problem D
Drunk Passenger

Time limit: 2 seconds
Memory limit: 1024 megabytes

Problem Description

Due to COVID-19, social distancing is applied in our daily life to prevent the spread of the disease. It changes our living styles a lot, especially the way of traveling. Now, many carriers cancel non-reserved seats and introduce seating rules to ensure that the distance between any two passengers is long enough.

On your trip to the 2022 ICPC World Finals, you take a flight. The airplane provides n reserved seats to n passengers. The passengers must queue up first, then they board the airplane one by one. You are the last passenger to board, since you are at the end of the queue. Unfortunately, the first passenger is drunk. The drunk passenger randomly goes to another passenger's seat and then sit there. You may assume the following.

1. The drunk passenger never takes their own seat.
2. The probability of any other seat taken by the drunk passenger is uniform.

Luckily, all the other passengers are not drunk. However, they don't want to move any passenger from a taken seat. If a passenger's seat has been taken by another passenger when boarding, the passenger would randomly take a vacant seat with equal probability. Otherwise, the passengers just take their own seats.

Please write a program to compute the probability that your seat is taken by another passenger.

Input Format

The input contains only one positive integer n.

Output Format

Output the probability that your seat is taken by another passenger. It is acceptable if the difference between your output and the answer is less then 10^{-6}.

Technical Specification

- $1 < n \leq 300$

Sample Input 1

```
2
```

Sample Output 1

```
1
```

Sample Input 2

```
3
```

Sample Output 2

```
0.75
```

Sample Input 3

```
4
```

Sample Output 3

```
0.66666666666666666666667
```

Note

The problem statement is a fiction.

Problem E
Eatcoin

Time limit: 5 seconds

Memory limit: 1024 megabytes

Problem Description

Eric developed a new algorithm to mine a cryptocurrency called Eatcoin. Since Eric's algorithm is an evalutionary algorithm, its performance keeps improving. On the d-th day of the execution of Eric's algorithm, it consumes p Eatcoins and then produces $q \times d^5$ Eatcoins where p and q are positive constants.

Eric wants to become a "duotrigintillionaire". A duotrigintillionaire is a person who has at least 10^{99} Eatcoins. Eric plans to exploit his algorithm to achieve his goal. Eric's algorithm can soon produce a huge amount of Eatcoins if he has enough Eatcoins. However, his algorithm cannot continue if he does not have p Eatcoins when needed.

Eric gives the values of p and q to you. Please write a program to help Eric to compute two numbers x and y defined as follows.

- x is the minimum number of Eatcoins required to execute Eric's algorithm to make him a duotrigintillionaire.
- y is the minimum number of days required to make Eric a duotrigintillionaire if Eric has exactly x Eatcoins before executing his algorithm.

Input Format

Two positive integers p and q are given in one line and separated by a space.

Output Format

Output two lines. Print x on the first line and y on the second line.

Technical Specification

- $1 \le q \le p \le 10^{18}$

Sample Input 1

```
50 1
```

Sample Output 1

```
117
42627697484524538
```

Sample Input 2

```
10 10
```

Sample Output 2

```
10
29041912218408574
```

Problem F
Flip

Time limit: 3 seconds
Memory limit: 1024 megabytes

Problem Description

Suppose you are given an array of n entries where each array entry is either 0 or 1. For any pair (ℓ, r) such that $1 \leq \ell \leq r \leq n$, $[a[\ell], a[\ell+1], \ldots, a[r]]$ is a subarray of the array $[a[1], a[2], \ldots, a[n]]$. An alternating subarray $[a[\ell], a[\ell+1], \ldots, a[r]]$ of $[a[1], a[2], \ldots, a[n]]$ if $a[\ell] \neq a[\ell+1] \neq \cdots \neq a[r]$. I.e., every entry in the subarray is different from its neighbors in the subarray. Since the definition of alternating subarrays only considers the entries in the subarrays, $[1, 0, 1]$ is still an alterating subarray of $[1, 1, 0, 1, 1]$.

In this problem, two types of operations will be applied on the given array.

- $1\ \ell\ r$: for every $i \in [\ell, r]$, change $a[i]$ into $1 - a[i]$.
- $2\ \ell\ r$: report the total number of pairs (x, y) such that $\ell \leq x \leq y \leq r$ and subarray $[a[x], a[x+1], \ldots, a[y]]$ is an alternating subarray.

Please write a program to maintain the given array. Your program must report the numbers efficiently.

Input Format

The first line contains two integers n and q indicating the length of the given array and the number of operations. The second line contain n space saparated numbers $a[1], a[2], \ldots, a[n]$ representing the given array $[a[1], a[2], \ldots, a[n]]$. Then q lines follows, and the i-th of them contains 3 integers t_i, ℓ_i, r_i where the i-th operation is $t_i\ \ell_i\ r_i$.

Output Format

For each operation of the second type, output the reported number on one line.

Technical Specification

- $1 \leq n \leq 200000$
- $1 \leq q \leq 200000$
- $a[i] \in \{0, 1\}$ for all $i \in \{1, 2, \ldots, n\}$.
- $t_j \in \{1, 2\}$ for all $j \in \{1, 2, \ldots, q\}$.
- $1 \leq \ell_j \leq r_j \leq q$ for all $j \in \{1, 2, \ldots, q\}$.

Sample Input 1

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

Sample Output 1

```
4
```

Sample Input 2

```
20 20
0 0 1 0 1 0 0 1 1 1 0 1 0 0 0 1 1 1 0 0
1 1 10
2 2 7
1 3 15
2 1 9
1 4 9
2 1 13
1 13 15
2 10 20
1 1 5
2 2 10
1 15 17
2 15 18
1 1 3
2 4 6
1 15 19
2 1 6
1 15 15
2 10 17
1 1 8
2 15 19
```

Sample Output 2

```
16
16
21
14
12
6
4
9
10
8
```

Problem G
Garden Park

Time limit: 3 seconds

Memory limit: 1024 megabytes

Problem Description

In the Garden park, there are n places of interest (numbered from 1 to n) and $n-1$ trails (numbered from 1 to $n-1$) connecting the places of interest. For every $i \in \{1, 2, \ldots, n-1\}$, trail i has two ends at place a_i and place b_i, and the trail does not pass any place of interest except its ends. Moreover, the trails do not have any intersection except the ends.

To protect the garden, visitors may only walk along the trails (in any direction) and inside the places of interest. For any pair of places of interest (x, y) where $x \neq y$, there exists a sequence of trails $s_1, s_2, \ldots, s_k$ satisfying the following conditions.

- Place x is an end of trail s_1.
- Place y is an end of trail s_k.
- For $1 \leq i < k$, trail s_i and trail s_{i+1} have a common end.
- If place z is the common end of trails s_i and s_{i+1} for some $i \in \{1, \ldots, k-1\}$, then z cannot be a common end of any other pairs of trails in $s_1, \ldots, s_k$.

In other words, a visitor move from x to y by walking along the trails $s_1, s_2, \ldots, s_k$ without visiting a place of interest twice. Such sequence is called a simple path from x to y.

The administration division of the park plans to host an event in the park. It puts labels on the trails. For trail t, the label on t is an integer $\ell(t)$, and a visitor can learn $\ell(t)$ by walk through trail t. A simple path $s_1, s_2, \ldots, s_k$ from x to y is with strictly increasing labels if $\ell(s_1) < \ell(s_2) < \cdots < \ell(s_k)$. By reporting m distinct simple paths with strictly increasing labels to the administration division, a visitor may win m free tickets for future visits.

You friend George just visit the park, and learn all labels on the trails. He wants to win free tickets for future visits with you. Please write a program to compute the number of distinct simple paths with strictly increasing labels in the garden park.

Input Format

The first line contains one integers n . The $(i+1)$-th line contains three integers a_i, b_i, c_i. Trail i connects place a_i and b_i, and the label $\ell(i)$ on trail i is c_i.

Output Format

Output the number of distinct simple paths with strictly increasing labels in the garden park.

Technical Specification

- $1 \leq n \leq 2 \times 10^5$
- $1 \leq a_i \leq n$ for $i \in \{1, 2, \ldots, n\}$.
- $1 \leq b_i \leq n$ for $i \in \{1, 2, \ldots, n\}$.
- $0 \leq c_i \leq 10^9$ for $i \in \{1, 2, \ldots, n\}$.
- $a_i \neq b_i$ for $i \in \{1, 2, \ldots, n\}$.

Sample Input 1

```
3
1 2 3
2 3 7
```

Sample Output 1

```
5
```

Sample Input 2

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

Sample Output 2

```
9
```

Problem H
A Hard Problem

Time limit: 3 seconds

Memory limit: 1024 megabytes

Problem Description

You are given a simple undirected graph consisting of n nodes and m edges. The nodes are numbered from 1 to n, and the edges are numbered from 1 to m. Node i has a non-negative integer value V_i and the weight $W_{u,v}$ of edge $\{u, v\}$ is defined as $\|V_u \oplus V_v\|$ where $\oplus$ is the exclusive-or operator (equivalent to `^` in C) and $\|x\|$ is the number of set bits in the binary representation non-negative integer z.

The node values $V_1, V_2, \ldots, V_n$ must satisfy q constraints. Each of the constraints can be represented as a 5-tuple (t, u, i, v, j).

- if $t = 0$, then $getBit(V_u, i) = getBit(V_v, j)$
- if $t = 1$, then $getBit(V_u, i) \neq getBit(V_v, j)$

where the function $getBit(x, i)$ returns the $(i + 1)$-th least significant bit of x. For examples, $getBit(11, 0)$ is 1 and $getBit(11, 2) = 0$. In the C programming language, $getBit(x, i)$ can be computed by `((x >> i) & 1U)` if x is a 32-bit unsigned integer and i is a non-negative integer at most 31.

Unfortunately, some node values are missing now. Your task is to assign new values to them to minimize $\sum_{\{u,v\}\in E} W_{u,v}$ without violating any given constraint. Please write a program to help yourself to complete this task.

Input Format

The input consists of five parts. The first part contains one line, and that line contains two positive integers n and m. n is the number of nodes, and m is the number of edges. The second part contains m lines. Each of them contains two integers u and v indicating an edge $\{u, v\}$ of the given graph. The third part contains one line. That line consists of n space-separated integers $x_1, x_2, \ldots, x_n$. If the node value V_i is missing, x_i will be `-1`. Otherwise, V_i is x_i. The fourth part contains one integer q indicating the number of constraints. The fifth part contains q lines, and each of them contains five space-separated integers t, u, i, v, j indicating that (t, u, i, v, j) is a constraint.

Output Format

Output an integer which is the minimum value under the q constraints. If it is not possible to satisfy all the constraints, output `-1`.

Technical Specification

- $1 \leq n \leq 1000$

- $1 \le m \le 5000$
- $-1 \le V_i < 2^{16}$
- $0 \le q \le 10$
- $t \in \{0, 1\}$
- $0 \le u, v < n$
- $0 \le i, j < 16$

Sample Input 1

```
4 4
1 3
1 2
3 2
0 3
-1 -1 60091 51514
2
1 2 0 1 5
0 2 6 0 15
```

Sample Output 1

```
13
```

Sample Input 2

```
3 2
0 1
1 2
-1 -1 -1
2
1 2 0 1 5
0 1 5 2 0
```

Sample Output 2

```
-1
```

Problem I
ICPC Kingdom

Time limit: 3 seconds
Memory limit: 1024 megabytes

Problem Description

The ICPC kingdom has n cities numbered from 1 to n, and there are m roads, numbered from 1 to m, connecting these cities. Each road connects two cities, and people can travel along the roads in both directions. There are a_i residents living in city i. Road j connects city u_j and city v_j. And the economic benefit of road j is $\left\lfloor \sqrt{a_{u_j} + a_{v_j}} \right\rfloor$.

One day, the enemy invaded the ICPC kingdom and destroys all roads. Fortunately, the ICPC troops defeated the enemy and stopped the invasion. To recover from the war, the king of ICPC hired w workers to fix the roads. Each worker can fix at most one road, and the i-th worker can only fix one of the roads whose indices are among $b_1, b_2, \ldots, b_{x_i}$.

Now the king wants to fix the roads to make the kingdom normal. Considering the cost, the king does not want the repair plan containing any useless road. If a repair plan contains a useless road, there exists a pair of city p and q such that there is more than one simple path from city p to city q. A simple path is a sequence of distinct roads $c_1, c_2, \ldots, c_z$ such that a trip along $c_1, c_2, \ldots, c_z$ will visit exactly $z + 1$ distinct cities.

The king asked you to calculate the maximum economic benefit if exactly k roads are fixed without useless roads. You need to calculate for each k between 1 and $n - 1$.

Input Format

The first line contains n and m seperated by a white space. n is the number of cities, and m is the number of roads. The second line contains n numbers $a_1, a_2, a_3 \ldots a_n$ seperated with white spaces indicating the number of the residents in cities $1, 2, \ldots, n$, respectively. There are m lines following, each line contains u_j and v_j separated with white space. Road j connects city u_j and city v_j. The third line contains a number w indicating the number of workers. The following w line indicates the roads that can be fixed the workers. The $(3 + i)$-th line contains several numbers separated by spaces. The first number is x_i indicating the number of roads that the i-th worker can fix. There are x_i distinct integers $b_1, b_2, \ldots, b_{x_i}$. following in that line. The i-th worker can only fix one of roads $b_1, b_2, \ldots, b_{x_i}$.

Output Format

Print $n - 1$ numbers where the i-th number is the the maximum economic benefit if the repair plan fixes exactly i roads without useless roads. If there is no such plan, you should print -1.

Technical Specification

- $1 \leq n \leq 100$

- $0 \le m \le 100$
- $1 \le a_i \le 10^9$
- $1 \le u_i, v_i \le n, u_i \ne v_i$
- $0 \le w \le 100$

Sample Input 1

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

Sample Output 1

```
2 3 4 -1
```

Problem J
JavaScript

Time limit: 2 seconds
Memory limit: 1024 megabytes

Problem Description

JavaScript is one of the most important computer languages now. It is high-level and multi-paradigm. It supports functional and imperative programming styles. However, the ICPC World Finals does not offer JavaScript for problem solving.

JavaScript is considered as a weakly typed language. It sometimes implicitly converts values of one type into another type. For example, the minus operator (`-`) does not have any meaning on strings, it is defined to operate on numbers. When the minus operator is applied on two strings, JavaScript will convert the operands from strings into numbers and then apply the minus operation. That is why `"2" + "2" - "2"` evaluates to `20` in JavaScript. Moreover, JavaScript converts a string into `NaN` (Not-a-Number) if the string does not represent a number. If any operand of a minus operation is `NaN`, then the result of the operation must be `NaN`. For example, `"a" + "2"` is `NaN`.

Given two strings `x` and `y`, please write a program to compute the result of `x - y` in JavaScript.

Input Format

There is only one line containing two space-separated non-empty strings `x` and `y`.

Output Format

Print the result of the minus operation (`x - y`) on one Line. If the result is an integer, please print it without the decimal point. If the result is not a number, please print `NaN`.

Technical Specification

- x and y consist of only English letters and digits.
- Both x and y have length less than 6.
- If x contains an English letter, then you may assume that JavaScript converts x into `NaN`.
- If y contains an English letter, then you may assume that JavaScript converts y into `NaN`.
- You may assume that the result is not a number if it is not an integer.

Sample Input 1

```
22 2
```

Sample Output 1

```
20
```

Sample Input 2

```
a 2
```

Sample Output 2

```
NaN
```

Sample Input 3

```
12345 a1a2a
```

Sample Output 3

```
NaN
```