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. Athelete 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 winner of the competitions, and silver medals are awarded to the runner-up. Most of the bronze medals are awarded to the second runner-up. However, there can be no second runner-up in some sports. The bronze medals are 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 integer 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.

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

```
4
0 0
1 0
1 1
0 1
```

Sample Output 1

```
3.14159265359
```

Sample Input 2

```
3
0 0
```

Sample Output 2

```
8.639379797371932
```

```
0 1
2 0
```

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 passenger 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

```
0
```

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
Mountain Park

Time limit: 5 seconds

Memory limit: 1024 megabytes

Problem Description

A binary string is a string consisting of only 0's and 1's. Elsa, Eric's boss, gave him a binary string s of length 20 and asked him to modify s into another binary string t within D days.

Eric really hates this task and therefore never modifies more than one character in a day. However, being forced to show Elsa the daily progress, Eric must modify some characters of the string every day. That means, the only possible way for Eric is to modify exact one character in each day before he finishes the task.

It is obviously cheating to have a character changed to something other than 0 and 1. Moreover, Eric will be caught cheating if the string is modified into the same binary string twice since Elsa has a good memory. That is, before the string is modified into t, all modifications result in unique strings. Note that Eric cannot modify the string into s which is the string given by Elsa, either.

Eric wants to spend as much time as possible. He is wondering if he can spend exact D days to have the string s modified into t. Please write a program to help Eric.

Input Format

The input contains three lines. The first line contains a binary string s. The second line contains a binary string t. The third line contains an integer D. Elsa asked Eric to modify the binary string s into t within D days.

Output Format

If there is no way to achieve what Eric wants, output `-1`. Otherwise, output D lines to represent one possible way. The i-th line contains a binary string, the result of the modification on the i-th day.

Technical Specification

- The strings s and t consist of only 0's and 1's.
- The length of s and the length of t are both 20.
- $1 \le D \le 500000$
- If there are multiple solutions, then you may output any of them.

Sample Input 1

```
00000000000000000000
11111111111111111111
5
```

Sample Output 1

```
-1
```

Sample Input 2

```
00000000001111111111
10000000001111111111
5
```

Sample Output 2

```
00000000001111111110
00000001001111111110
10000001001111111110
10000000001111111110
10000000001111111111
```

Problem F
F

Time limit: 3 seconds

Memory limit: 1024 megabytes

Problem Description

Suppose you are given an array of n entries where each entry of the array is either 0 or 1. A subarray $[a[\ell], a[\ell+1], \dots, a[r]]$ of an array $[a[1], a[2], \dots, a[n]]$ if $1 \le \ell \le r \le n$. An alternating subarray $[a[\ell], a[\ell+1], \dots, a[r]]$ of $[a[1], a[2], \dots, a[n]]$ if $a[\ell] \ne a[\ell+1] \ne \dots \ne a[r]$. I.e., an arbitrary entry in the subarray is different from its neighbors in the subarray.

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 number of pairs (x, y) such that $\ell \le x \le y \le r$ and subarray $[a[x], a[x+1], \dots, a[y]]$ 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 $a[1], a[2], \dots, a[n]$ representing the given array $[a[1], a[2], \dots, a[n]]$. 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

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

Sample Input 1

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

Sample Output 1

```
6
```

Almost blank page

Problem G

G

Time limit: 3 seconds

Memory limit: 1024 megabytes

Problem Description

In a forest 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 forest, 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 forest 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 forest 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 3
*.*
...
*.*
```

Sample Output 1

```
4
*C*
C.C
*C*
```

Sample Input 2

```
2 4
*..*
....
```

Sample Output 2

```
3
*C.*
C.C.
```

Almost blank page

Problem H

H

Time limit: 3 seconds

Memory limit: 1024 megabytes

Problem Description

There's no such thing as public opinion.

Jordan Ellenberg, American Mathematician

In K City lives n residents who want to build a connection network with each other. However, some residents want the network wires colored black while the others want the wires colored white. The opinion of resident i can be quantified as a number a_i. If we build a network wire between residents i and j, the cost of this wire will be $a_i \times a_j$.

The mayor of K City wants to build a network such that:

1. There are exactly $n - 1$ wires used.
2. For any two different residents i and j, there exists a sequence $p_1, \cdots, p_k$ such that $p_1 = i$, $p_k = j$ and residents p_ℓ and $p_{\ell+1}$ share a wire for $1 \leq \ell < k$.

In other words, the network should be a tree.

You, the renowned mathematician of K City, want to know not only the *minimum* cost to build the network. In the name of confusion, you also want to know the *maximum* cost!

Input Format

The first line begins with a number n indicating the number of residents. The second line contains n numbers $a_1, a_2, \ldots, a_n$. The opinion of resident i is the quantified as a_i.

Output Format

Output two numbers separated by a blank in a line. The numbers are the *minimum* cost and the *maximum* cost to build the network, respectively. Since the absolute value of the costs may be extremely large, you have to modulo the answer with $10^9 + 7$. Please note that the modulo of a number (defined by Donald Knuth) is $a \bmod b = a - b\lfloor \frac{a}{b} \rfloor$. The output number should be non-negetive.

Technical Specification

- $1 \leq n \leq 10^6$
- $|a_i| \leq 10^6$

Sample Input 1

```
10
-5 -10 -7 -7 -3 -1 -7 -5 -8 -6
```

Sample Output 1

```
58 490
```

Sample Input 2

```
10
-5 1 2 -2 -1 1 -5 5 -10 6
```

Sample Output 2

```
999999779 183
```

Sample Input 3

```
10
0 0 0 0 0 0 0 0 0 0
```

Sample Output 3

```
0 0
```

Sample Input 4

```
10
10 8 9 3 8 8 0 5 3 10
```

Sample Output 4

```
0 540
```

Problem I
I

Time limit: 3 seconds

Memory limit: 1024 megabytes

Problem Description

Teams from variaous universities compete in ICPC regional contests for tickets to the ICPC World Finals. The number of tickets allocated to every regional contest may be different. The allocation method in our super region, Asia Pacific, is based on a parameter called site score.

Site scores will only count teams and universities solving at least one problem, in the regional contest or its preliminary contest TOPC. In 2020, the formula for calculating the site score of the Taipei-Hsinchu regional contest is much simpler than past years. Let

- U_R be the number of universities solving at least one problem in the regional contest.
- T_R be the number of teams solving at least one problem in the regional contest.
- U_O be the number of universities solving at least one problem in TOPC.
- T_O be the number of teams solving at least one problem in TOPC.

The site score of 2020 Taipei-Hsinchu regional contest will be $56U_R + 24T_R + 14U_O + 6T_O$. Please write a program to compute the site score of the 2020 Taipei-Hsinchu regional contest.

Input Format

The input has only one line containing four blank-separated positive integers U_R, T_R, U_O, and T_O.

Output Format

Output the site score of the 2020 Taipei-Hsinchu regional contest.

Technical Specification

- $0 < U_R \leq T_R \leq 120$
- $0 < U_O \leq T_O \leq 1000$

Sample Input 1

```
1 1 1 1
```

Sample Output 1

```
100
```

Sample Input 2

```
1 10 100 1000
```

Sample Output 2

```
7696
```

Note

The problem statement is fiction. The real site score has a different formula.

Almost blank page

Problem J
J

Time limit: 3 seconds
Memory limit: 1024 megabytes

Problem Description

Alex is attending the first edition of Robotic World Championship of Table Tennis. A competition that have all of the matches having the same rules listed below:

- A match shall consist of the best of 7 games, i.e., the results of matches must be 4 games to k, where $0 \leq k \leq 3$.
- A game shall be won by the player first scoring 11 points unless both players score 10 points, when the game shall be won by the first player subsequently gaining a lead of 2 points. For example, a game can be won at scores like 11-5, 11-9 or 12-10, but not 10-5 or 11-10.
- After each 2 points have been scored the receiving player shall become the serving player and so on until the end of the game, unless both players score 10 points, when the sequences of serving and receiving shall be the same but each player shall serve for only 1 point in turn. That is, the servicing order of the first 20 points is `AABBAABBAABBAABBAABB`, and will be followed by `ABABAB`... if necessary.
- The player serving first in a game shall receive first in the next game of the match.

Experience tells that when two robots clashes into each other, the variances affecting their winning chances can be simplified to who's serving for the point. This is due to the performances of the robots are physically consistent and won't be affected mentally.

Alex have listed some of the possible matchups, simplified to the winning chance of each servicing point of the robots, for you. Now it is your job to help him calculate the winning chance of each match for them.

Input Format

The first line of the input consists of a single number T, indicating that there will be T test cases following.

Each of the following test case consists of two space-separated real numbers P_A and P_B in one line, where P_A denotes the Robot A's chance of winning the point when A is serving and P_B denotes the Robot B's chance of winning the point when B is serving.

The Robot A always serves first in the very first game of the match.

Output Format

For each test case, output one real number in one line: the winning chance of A.

Technical Specification

- $T \leq 100$
- $0 \leq P_A \leq 1$ and has at most 2 digits after the decimal point in the input.
- $0 \leq P_B \leq 1$ and has at most 2 digits after the decimal point in the input.
- $0 < P_A + P_B < 2$
- The answer will be considered correct if it is within an absolute error of 10^{-8} of the correct answer.

Sample Input 1

```
3
1 0
0.5 0.5
0.00 1.00
```

Sample Output 1

```
1
0.5
0.000000000
```

References

The rules are revised from the Chapter 2 “The Laws of Table Tennis” of The International Table Tennis Federation (ITTF) Handbook 2020.