Problem A

Convert Floating-point Numbers into Fractions

Time limit: 1 second

Write a program to convert floating-point numbers into fractions. For example, on input 3.14 print out 157/50.

Input Format

The input file may contain many test data. Each test data contains a positive floating-point number stored in a line. Note that a whole number without decimal point, such as 3, can also be a valid input data. The maximum number of digits of the floating-point number is 16, not including a possible decimal point. The last line of the input file contains 0. Your program must exit if the input data is 0.

Output Format

For each test data x print a fraction number p/q whose value is equal to x. Note that p and q must be relatively prime, i. e., $\gcd(p, q) = 1$. For example, 4/6 is not allowed.

Examples

Input	Output
3 3.14 1.234567890123456 0	3/1 157/50 19290123283179/15625000000000

Problem B

Square Matrix Traversal

(Time Limit: 3 seconds)

Consider an $n \times n$ square matrix M with n^2 elements which are denoted as $m_{i,j}$'s for $1 \leq i, j \leq n$. Give a traversal sequence of all elements in the matrix M shown below:

$$m_{1,1}, m_{1,2}, m_{2,2}, m_{2,1}, m_{3,1}, m_{3,2}, m_{3,3}, m_{2,3}, \dots, m_{n,1} (\text{or } m_{1,n})$$

This sequence starts at $m_{1,1}$ and ends at $m_{n,1}$ ($m_{1,n}$) if n is an even (odd) number. Take a look at the following 5×5 square matrix M and its corresponding Traversal Ordering Matrix (TOM).

$$\begin{bmatrix} m_{1,1} & m_{1,2} & m_{1,3} & m_{1,4} & m_{1,5} \\ m_{2,1} & m_{2,2} & m_{2,3} & m_{2,4} & m_{2,5} \\ m_{3,1} & m_{3,2} & m_{3,3} & m_{3,4} & m_{3,5} \\ m_{4,1} & m_{4,2} & m_{4,3} & m_{4,4} & m_{4,5} \\ m_{5,1} & m_{5,2} & m_{5,3} & m_{5,4} & m_{5,5} \end{bmatrix} \qquad \begin{bmatrix} 1 & 2 & 9 & 10 & 25 \\ 4 & 3 & 8 & 11 & 24 \\ 5 & 6 & 7 & 12 & 23 \\ 16 & 15 & 14 & 13 & 22 \\ 17 & 18 & 19 & 20 & 21 \end{bmatrix}$$

(a) 5×5 square matrix M (b) Traversal Ordering Matrix

For each number in TOM, it indicates the order of traversing $m_{i,j}$ in M. For example, the traversing orders of $m_{3,3}$ and $m_{1,5}$ are 7 and 25, respectively.

Input Format

The input starts with an integer *t* indicating the number of test cases. For each test case, it has one line of input. It has two integers *i* and *j* representing the element $m_{i,j}$ in the square matrix M.

Output Format

For each test case, there is one line of output. It outputs an integer *k* indicating the order of $m_{i,j}$ in the traversal sequence.

Technical Specification

- t is an integer and $0 < t \leq 100$
- i, j are integers and $0 < i, j \leq 2^{32} - 1$

Problem B

Example

Sample Input:	Sample Output:
4 1 1 2 1 1 5 4 4	1 4 25 13

Problem C
The Fraction Problem

(Time Limit: 5 seconds)

Given an integer $k(> 0)$, we can always find three positive finite integers x, y, and z, $x \geq y \geq z > 0$, satisfy

$$\frac{1}{k} = \frac{1}{x} + \frac{1}{y} + \frac{1}{z}$$

For example, given $k = 1$, we can find 3 triplets $< x, y, z > = < 6, 3, 2 >$, $< 4, 4, 2 >$, or $< 3, 3, 3 >$, to satisfy the above equation (see below).

$$\frac{1}{1} = \frac{1}{6} + \frac{1}{3} + \frac{1}{2}$$

$$\frac{1}{1} = \frac{1}{4} + \frac{1}{4} + \frac{1}{2}$$

$$\frac{1}{1} = \frac{1}{3} + \frac{1}{3} + \frac{1}{3}$$

Write a program to count how many such triplets $< x, y, z >$ there are for any given k?

Input Format

The first line is an integer T indicating the number of test cases. The following lines denote each test case. Each test case contains one line of an integer, which is the value of k $(0 < k \leq 2000)$.

Output Format

For each test case, output one line of an integer which is the number of the corresponding $< x, y, z >$ triplets for the given k.

Problem C

Technical Specification

- T is an integer. $0 < T \leq 10$.
- k is an integer. $0 < k \leq 2000$.
- x, y, z are positive finite integers. $x \geq y \geq z > 0$.

Example

Sample Input:	Sample Output:
3 1 2 3	3 10 21

Problem D

Problem: Noise in Carriage

(Time Limit: 5 seconds)

Taipei Metro, also known as the MRT or formally as the Taipei Mass Rapid Transit System, is a rapid transit system serving metropolitan in Taipei, Taiwan. In the future, more and more lines will be constructed in Taipei MRT system. Therefore, the routing path of Taipei MRT will become very complicated. MRT system brings citizen much convenience, but it also causes a new problem, noise pollution in carriages. A decibel (dB) is a unit of measurement that indicates how loud a sound is. Results from experimental trial testing shows that the sound human can hear has sound intensity level from 0 to 130 decibels and the higher sound intensity level is considered painful. Therefore, to understand the influence of the carriage noise made by Taipei MRT, Taipei city government has a project to find the noise ceiling of each two stations in Taipei MRT system. Take from station **Zhongshan** to station **Dongmen** for example. You may route though many different paths between these two stations and two of routing paths are presented in Figure 1. One of routing path is ❶: **Zhongshan → Beimen → Ximen → Xiaonanmen → Chiang Kai-Shek Memorial Hall → Dongmen** and the other one is ❷: **Zhongshan → Songjiang Nanjing → Zhongxiao Xinsheng → Dongmen**. It is observed that a passenger must tolerate 115 and 100 decibels of noise intensity level with routing through path ❶ and path ❷, respectively. This means that the passenger will feel more comfortable while routing through path ❷. Of course, there are other routing paths, too. We all know that you are a good programmer and Taipei city government will hire you to write a program for handling this project.

Input Format

The input may contain multiple test cases. The first line of each test case contains three integers $\boldsymbol{S}(\leq 300)$, $\boldsymbol{C}(\leq 10000)$ and $\boldsymbol{Q}(\leq 1000)$, where $\boldsymbol{S}$ indicates the number of stations (numbered using distinct integers ranging from 1 to $\boldsymbol{S}$), C represents the number of connection lines and $\boldsymbol{Q}$ is the number of queries. Each of the next C lines contains three integers: $\boldsymbol{a}$, $\boldsymbol{b}$ and $\boldsymbol{d}$ indicating that the average noise intensity level on the carriage between the stations $\boldsymbol{a}$ and $\boldsymbol{b}$ ($\boldsymbol{a} \neq \boldsymbol{b}$) is $\boldsymbol{d}$ decibels. Each of the next $\boldsymbol{Q}$ lines contains two integers a and b ($\boldsymbol{a} \neq \boldsymbol{b}$) asking for the minimum noise intensity level a passenger must tolerate in order to take Taipei MRT from station $\boldsymbol{a}$ to station $\boldsymbol{b}$. The input will terminate with three zeros form $\boldsymbol{S}$, $\boldsymbol{C}$ and $\boldsymbol{Q}$.

Problem D

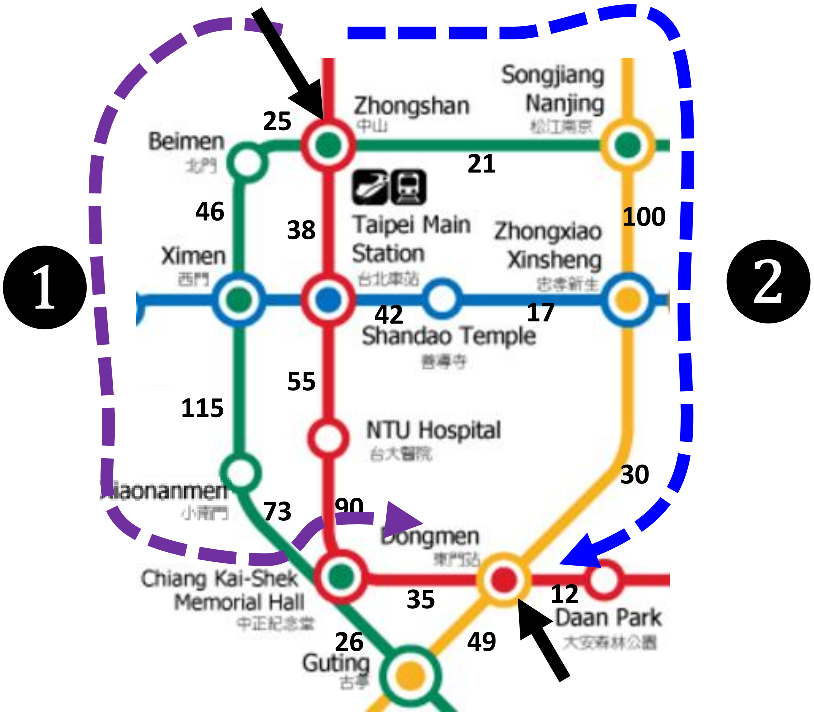


Figure 1. Two of routing paths from **Zhongshan** station to **Dongmen** station.

Output Format

For each test case, you must print the following messages:
Case #n:

Here n stands for the case number (starting from 1). Then for each query in the input print a line giving the minimum noise intensity level (in decibels) a passenger must tolerate in order to take Taipei MRT from ***a*** station to ***b*** station. If there exists no routing path between these two stations and then just print the line "no path". Print a blank line between two consecutive test cases, including the last test case.

Problem D

Example

Sample Input:	Sample Output:
7 9 3 1 2 50 1 3 60 2 4 120 2 5 90 3 6 50 4 6 80 4 7 70 5 7 40 6 7 140 1 7 2 6 6 2 7 6 3 1 2 50 1 3 60 2 4 120 3 6 50 4 6 80 5 7 40 7 5 1 7 2 4 0 0 0	Case #1 80 60 60 Case #2 40 no path 80

Problem E
Range Minimum Queries

(Time Limit: 2 seconds)

Denote the elements of a length-n integer A by $A[1]$, $A[2]$, ..., $A[n]$. Bob is given n pairs of integers, denoted (i_1, j_1), (i_2, j_2), ..., (i_n, j_n), such that $1 \le i_k \le j_k \le n$ for all $k \in \{1,2,\ldots,n\}$. He wants to find the minimum among $A[i_k]$, $A[i_k + 1]$, ..., $A[j_k]$ for each $k \in \{1,2,\ldots,n\}$. Please help him.

For example, let $n = 4$, $A[1] = 8$, $A[2] = 6$, $A[3] = 2$, $A[4] = 9$, $i_1 = 2$, $j_1 = 4$, $i_2 = j_2 = 2$, $i_3 = 1$, $j_3 = 2$, $i_4 = 3$ and $j_4 = 4$. Then

$$\min\{A[i_1], A[i_1 + 1], \ldots, A[j_1]\} = 2,$$
$$\min\{A[i_2], A[i_2 + 1], \ldots, A[j_2]\} = 6,$$
$$\min\{A[i_3], A[i_3 + 1], \ldots, A[j_3]\} = 6,$$
$$\min\{A[i_4], A[i_4 + 1], \ldots, A[j_4]\} = 2.$$

Input Format

The first line of the input is n. The second line consists of $A[1]$, $A[2]$, ..., $A[n]$. For each $k \in \{1,2,\ldots,n\}$, the $(k+2)$nd line consists of i_k and j_k. Two consecutive numbers in a line are separated by a space.

Output Format

For each $k \in \{1,2,\ldots,n\}$, the kth line of the output shall be the minimum among $A[i_k]$, $A[i_k + 1]$, ..., $A[j_k]$.

Technical Specification

- $n \le 100000$.
- $1 \le A[i] \le 1000000$ for all $i \in \{1,2,\ldots,n\}$.
- $1 \le i_k \le j_k \le n$ for all $k \in \{1,2,\ldots,n\}$.

Problem E

Example

Sample Input:	Sample Output:
6 3 1 2 6 5 4 4 6 2 4 3 6 1 6 2 5 3 3	4 1 2 1 1 2

Problem F
Battle of Pokémons

(Time Limit: 3 seconds)

In a Pokémon game, GYM is a place where a Pokémon can reside if it wins a battle. If you want to take over a GYM, you must beat the Pokémon who occupies the GYM currently. Each occupier has a *CP* (Combat Point) value, and the value is decreased after it is attacked. You cannot take over a GYM until the occupier's *CP* value is equal to or less than zero. In this game, you can attack an occupier in several rounds. In each round, you can launch an attack containing an attack power (*AP*), and the occupier's *CP* value is decreased accordingly. For example, in the first round, the occupier's *CP* value is 5, and after an attack it becomes 2 as the attack has 3 attack power. However, if you cannot beat the occupier in the current round, its *CP* value will increase in the next round. For example, the occupier's *CP* value becomes 3 by adding 1 recovery point (*RP*) in the second round. Besides, there is a fatigue factor (*FF*) of an attack, which means that in each successive round the attack power $FF \times AP$ less than it has in the previous round. For example, suppose that the fatigue factor is 5%. Your attack power becomes $3 - 3 \times 5\% = 2.85$ in the second round.

The following table shows an example of a battle process, indicating you can take over the GYM in Round 3.

	Occupier's CP	Attack Power	Occupier's CP after an attack
Round 1	5	3	2
Round 2	3	2.85	0.15
Round 3	1.15	2.7075	-1.5575

You have to solve this problem in general. Based on the given values of occupier's CP value, attack power, recovery point and fatigue factor, you must determine whether you can win the battle first. If yes, then you must find out in which round you can win the battle.

Problem F

Input Format

The first line contains an integer N, indicating the number of test cases. Each of the following N lines contains four numbers corresponding to the input CP (indicating occupier's CP value), the input AP (indicating attack power), the input RP (indicating recovery point), and the input FF (indicating fatigue factor expressed as a percentage) in each of the N cases.

Output Format

For each test case, output a line indicating whether you can win the battle, and in which round if you win.

Technical Specification

- CP is a float-point number. $1 \leq CP \leq 100$.
- AP is a float-point number. $1 \leq AP \leq 100$.
- RP is a float-point number. $1 \leq RP \leq 100$.
- FF is a float-point number. $1 \leq FF \leq 100$.

Example

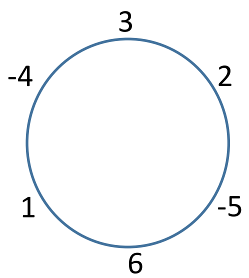
Sample Input:	Sample Output:
4 5 3 1 5 5 2 2 5 5 3 1 50 50 6 3 1	Win in Round 3. Fail to win the battle. Fail to win the battle. Win in Round 19.

Problem G

Maximum Sum Circular Subarray

(Time Limit: 3 seconds)

An array A of n elements $A[0]$, $A[1]$, …, $A[n\text{-}1]$ is circular if its first element $A[0]$ is adjacent to its last element $A[n\text{-}1]$. Given a circular array A of n integers, your task is to write a program that calculates the largest sum of consecutive elements in this array. The value of each array element is either positive or negative, and the absolute value of each array element is no more than 100,000. For example, there is a 6-element circular array with values 3, -4, 1, 6, -5, and 2 (as shown below). Note that the values 2 and 3 in this circular array are adjacent.



The largest sum of consecutive elements in this circular array is 8 which is the sum of 2, 3, -4, 1, and 6.

Input Format

The first line is an integer T indicating the number of test cases. The following lines denote each test case. Each test case is composed of two lines. The first line of each test case is an integer n which indicates the number of elements in the circular array A, and the second line contains n integers which are the values of array elements $A[0]$, $A[1]$, …, $A[n\text{-}1]$.

Output Format

For each test case, output one line of an integer which is the largest sum of consecutive elements in this circular array A.

Technical Specification

- T is an integer. $1 \le T \le 20$.
- n is an integer. $1 \le n \le 100{,}000$.

Problem G

Example

Sample Input:	Sample Output:
3 3 -1 2 -3 6 3 -4 1 6 -5 2 5 1 2 3 4 5	2 8 15

Problem H
Lottery

(Time Limit: 3 seconds)

A special design of Lottery called Lotto n/m is held. Players choose n distinct numbers from one to m per bet. There are also n numbers drawn by the held organization. If players' n chosen numbers match all the n drawn numbers, then they share the first prize. For example, Lotto **6**/**49** is held in Taiwan.

Alice prepares to place w bets on Lotto n/m for every draw. There are $C(m,n)=m!/n!/(m-n)!$ combinations. Each combination is expressed from smaller to larger. Alice sorts all the combinations of n distinct numbers by the lexicographic ordering. Alice picks a number p which is coprime to $C(m,n)$ as her lucky number and calculate a number q such that $(p*q)\%C(m,n)=1$ where % is the modulo operator (or the remainder operator). Alice first strikes out the first combination **1-2-3-…-n** and then strikes out one combination every q ones cyclically until there are exact w remained ones. That is, Alice sequentially strikes out $(q*k\%C(m,n)+1)$–th combination for k=**0,1,2,3,…** until there are exact w remained ones. Then Alice places **1** bets on each of w combinations. Now, given n, m, p, w, can you help Alice to calculate these w combinations?

A Lotto n/m combination $a_1 - a_2 - a_3 - \cdots - a_n$ is less than $b_1 - b_2 - b_3 - \cdots - b_n$ in lexicographic ordering if there exists a $1 \le k \le n$ such that $\forall_{i<k}\, a_i = b_i$ and $a_k < b_k$.

Take n=**3**, m=**5**, p=**7**, w=**4** as an example. There are C(**5**,**3**)=**10** combinations. They are **1-2-3**, **1-2-4**, **1-2-5**, **1-3-4**, **1-3-5**, **1-4-5**, **2-3-4**, **2-3-5**, **2-4-5**, and **3-4-5** in the lexicographic ordering. We also have q=**3** because (**7*****3**)%**10**=**1**. Alice first strike out **1-2-3**, and then **1-3-4**, **2-3-4**, **3-4-5**, **1-2-5**, and **1-4-5**. Finally, **4** combinations **2-4-5**, **1-2-4**, **1-3-5**, and **2-3-5** are kept. Output them in lexical graphic ordering as **1-2-4**, **1-3-5**, **2-3-5**, and **2-4-5**.

Input Format

The first line is a number indicating the number of test cases. Each test case has **4** numbers n, m, p, w in one line.

Output Format

For each test case, output "Tickets:" in the first line, and then output the w combinations by lexical graphic ordering. For each number in each combination, please add one leading zero if the number is less than **10**. That is, for combination **1-2-3**, please output **01-02-03**.

Problem H

Technical Specification

- There are at most **10** test cases.
- $\mathbf{1 \le m \le 75}$.
- $\mathbf{1 \le n \le 20}$.
- $\mathbf{1 \le p, q \le C(m, n)}$**-1**.
- $\mathbf{1 \le w \le 10}$.

Example

Sample Input:	Sample Output:
2 3 5 7 4 6 49 13 3	Tickets: 01-02-04 01-03-05 02-03-05 02-04-05 Tickets: 01-10-13-26-29-47 06-14-18-28-30-33 07-20-24-29-38-45