Problem A. Birthday Cake

Input file:	`standard input`
Output file:	`standard output`
Time limit:	1 second
Memory limit:	256 megabytes

Meow's birthday is coming. To celebrate his birthday, his friends bought him a rectangular cake with sides X and Y.

Now including Meow, there are a total of N individuals to share this big cake, and everyone must get the cake with the **same area**.

Meow will be the one in charge of cutting the cake, but he has to follow a particular rule. He can make a cut only if the cut is parallel to one side (any side) of the cake, and the cake must be cut into two pieces.

Thus, to slice the cake into N pieces, Meow must slice $N - 1$ times.

In order to make each piece of cake look beautiful, Meow would like the maximum value for the ratio of the long side to the short side of all those N pieces to be the smallest.

Can you help Meow find this ratio?

Input

The input contains 3 integer – X, Y, N ($1 \le X, Y \le 10^4, 1 \le N \le 10$) – the length of the cake, the width of the cake, and the number of pieces the cake should be cut into.

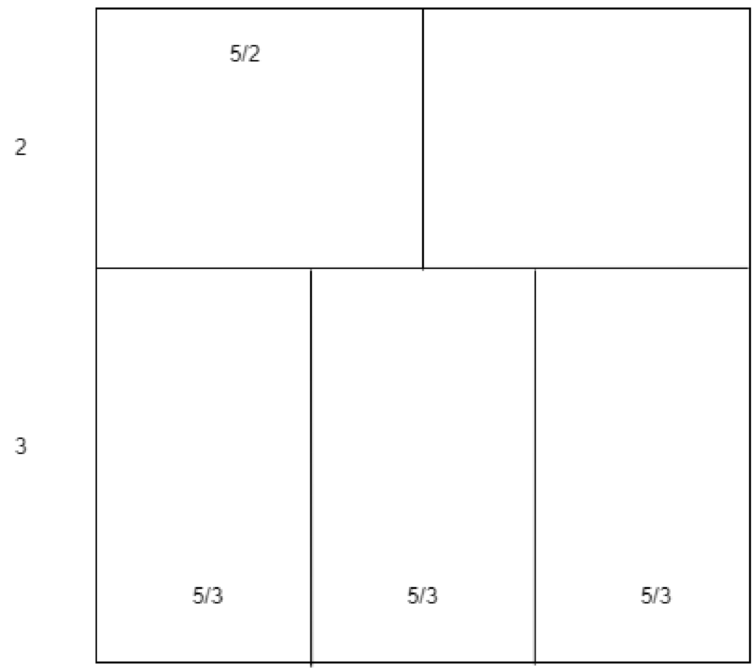
Output

Print a single float number, the ratio, with accuracy up to 6 decimal places, rounded off.

Example

standard input	standard output
5 5 5	1.800000

Note



Problem B. Catnip Trading

Input file:	`standard input`
Output file:	`standard output`
Time limit:	1 second
Memory limit:	256 megabytes

Meow and his friends are obsessed with catnip. They all live on the same street and they decided to buy or sell some catnip one day. Surprisingly, demand and supply is always the same so that everyone gets what they want.

Unfortunately, transporting catnip from one house to another requires energy. Cats are known to be lazy creatures, they want to minimise their overall effort in transporting catnip from one house to another. Fortunately, all catnip are equally as good, they do not care where the catnip is from.

Meow and his friends lives along a straight line with equal distance between adjacent houses. Transporting one unit of catnip from one house to another adjacent house result in 1 unit of energy.

What is the minimum amount of energy needed to transport the catnips such that every cat has its demand fulfilled?

Input

The first line consists of the number of test cases, t ($1 \leq t \leq 100$)

Each test case starts with the number of residents on that street, n ($2 \leq n \leq 10^5$).

The following line will have n integers. For each a_i ($-1000 \leq a_i \leq 1000$), if $a_i > 0$, the resident in house i wants to buy a_i number of catnip; else if $a_i < 0$, the resident in house i wants to sell a_i number of catnip. You may assume that $\sum_{i=1}^{n} a_i = 0$.

Output

For each test case, print the minimum amount of energy needed so that every cat has its demand fulfilled.

Example

standard input	standard output
2 5 5 -4 1 -3 1 6 -100 -100 -100 100 100 100	9 900

Problem C. Checkers

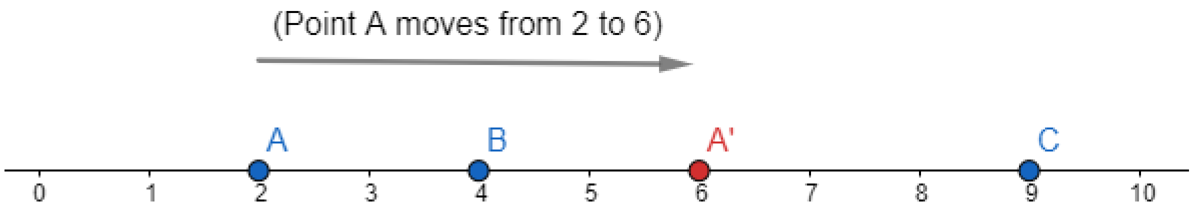
Input file:	`standard input`
Output file:	`standard output`
Time limit:	1 second
Memory limit:	256 megabytes

Checkers

Checkers, also known as draughts, is a group of strategy board games for two players which involve diagonal moves of uniform game pieces and mandatory captures by jumping over opponent pieces.

In this question we will consider a simplified version of the checkers game.

Assume there is a 1-D number line. Pieces can only be placed on the integers points. No more than one piece can be placed on each point.



Schematic for Simplified Checkers game

Let's play a simple game with checkers: assume there are 3 pieces on the board, in positions A, B and C. We want to move their positions to X, Y, Z with minimal 'jumping' required. (The pieces are indistinguishable)

The rules of 'jumping' are very simple:

1. Choose any piece
2. Select another piece to be your central axis.
3. Move the piece you choose to 'jump' over the central axis piece.
4. After the 'jump', the distance between the two pieces should remain the same.
5. During the whole process of 'jumping', you can 'jump' over one piece only, any move that involves jumping over more than one piece is prohibited.
6. No two pieces are allowed to be in a single point.

For example, point A is at $x = 2$, the central axis piece chosen is point B at $x = 4$. Thus, the distance is 2. Point A will moves from point 2 to point 6.

Can you write a program, first determine whether the task can be completed. If possible, output the minimum number of 'jumping' required.

Input

The first line contains 3 integers , A, B, C $(-10^8 \leq A, B, C \leq 10^8)$. A, B, C are distinct. $(A \neq B \neq C)$

The second line contains 3 integers , X, Y, Z $(-10^8 \leq X, Y, Z \leq 10^8)$. X, Y, Z are distinct. $(X \neq Y \neq Z)$

Output

If there is no solution, print "NO".

If it is possible to reach, print "YES"in the first line. Then print the minimum 'jumping' required in the second line.

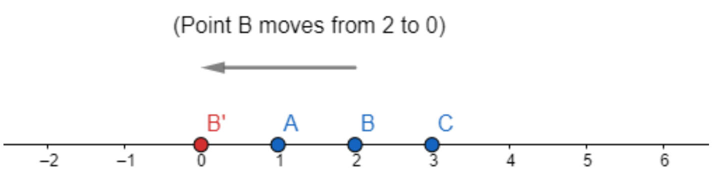
Examples

standard input	standard output
1 2 3 0 3 5	YES 2
1 5 10 -1 3 7	NO

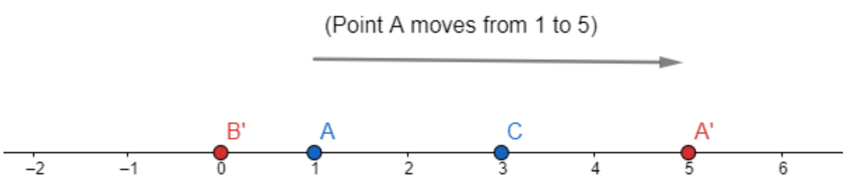
Note

The first example is illustrated as below.

1.



2.



(Note that the points are indistinguishable. A,B,C are labelled only for illustration purpose)

Problem D. Chocolate Buffet

Input file:	`standard input`
Output file:	`standard output`
Time limit:	1 second
Memory limit:	256 megabytes

It is a well known fact that Meow loves chocolate. His friends, Miau brought him many chocolates, each with different flavour and sizes. The weight of these chocolate (in grams) are $1, 2, 4, 8, 16, ...$ (Every subsequent weight of the chocolate is double to the prior weight, i.e. $n = \{2^n \mid n \in \mathbb{N}_0\}$ and there is only one chocolate for each weight)

Meow loves to eat as many chocolate as possible, and he needs to eat at least X grams of chocolate to be satisfied. However, since Meow is afraid to be a fat cat, he cannot eat more than Y grams (eating up to Y grams is fine).

Meow wishes to know what is the maximum number of chocolate he can eat.

Input

The first line contains an integer t $(1 \le t \le 1000)$ – the number of test cases.

Each test case contains 2 integers, X and Y $(1 \le X \le Y \le 2^{63} - 1)$

Output

Print a single integer – the maximum number of chocolate Meow can eat.

Example

standard input	standard output
1 16 25	4

Note

For the first example, Meow can eat maximum 4 chocolates : 1, 2, 4, 16

(there might exist more than 1 possible way to eat)

Problem E. Count Equal Pairs

Input file:	`standard input`
Output file:	`standard output`
Time limit:	1 second
Memory limit:	256 megabytes

Meow is given two arrays, represented by a and b. The elements in both arrays are sorted in non-decreasing order. How many number of pairs (i, j) Meow can find from the two arrays where $a_i = b_j$ $(0 \le i < n;\ 0 \le j < m)$

Input

The first line contains an integer t $(1 \le t \le 100)$ - the number of test cases.

The first line of each test case contains two integers n and m, representing the size of the arrays $(1 \le n, m \le 10^5)$.

The second line of each test case contain n integers a_i, where $(-10^3 \le a_i \le 10^3)$.

The third line of each test case contain m integers b_j, where $(-10^3 \le b_j \le 10^3)$.

Output

For every test case, print the number of **equal** pairs that can be formed from the two arrays.

Example

standard input	standard output
2 8 7 1 1 3 3 3 5 8 8 1 3 3 4 5 5 5 3 4 3 4 5 3 4 4 5	11 4

Note

In second example, there are 4 such pairs – (3,3),(4,4),(4,4),(5,5)

Problem F. Dice

Input file:	`standard input`
Output file:	`standard output`
Time limit:	1 second
Memory limit:	256 megabytes

There are n people standing in a queue, and Meow is standing on the m^{th} place.

Every round, the person in front of the queue gets to throw a fair dice (1-6).

- If the number rolled is 1, the person in front of the queue wins.
- If the number rolled is $2, 3, 4$, the person in front of the queue will have to go the the back of the queue.
- If the number rolled is $5, 6$, the person in front of the queue will get kicked out from the queue.

If the queue is only left with 1 person, that person automatically wins. What is the probability of Meow wins?

Input

The input contains two integers, n, m ($1 \leq m \leq n \leq 1000$) – the number of people in the queue and the position of Meow in the queue (starting from 1).

Output

Print a real number representing the probability of winning, output your answer with accuracy up to 9 decimal places, rounded off. (Your answer will be accepted if it is within 1×10^{-9} of the judge's answer)

Example

standard input	standard output
`2 1`	`0.444444444`

Problem G. Railway Designer

Input file:	`standard input`
Output file:	`standard output`
Time limit:	1 second
Memory limit:	256 megabytes

You are working as a railway designer for a company who have n railways stations around the globe. Currently, each station are fully connected to each other by a railway, which means a train can travel in both direction from one station to another.

Each railway have a maintenance cost which is equal to the geographical distance between two stations in kilometer (km). Your manager tells you that cycles are not allowed in the railway connections.

Since the current railway connections incur a high maintenance cost and they need to redesign their railway connections to cut cost, how can you minimize the total number of railways that connect the stations such that the train can still travel to every station?

Find the **minimum** total maintenance cost for the new railways connection you designed and output your answer as integer after applying **floor function**.

You may calculate the maintenance cost with Haversine formula:

$$d = 2r \arcsin\left(\sqrt{\sin^2\left(\frac{\phi_2 - \phi_1}{2}\right) + \cos(\phi_1)\cos(\phi_2)\sin^2\left(\frac{\lambda_2 - \lambda_1}{2}\right)}\right)$$

In the calculation, assume r = 6371.

ϕ_1 is the latitude of the first station and ϕ_2 is the latitude of second station in radian.

λ_1 is the longitude of the first station and λ_2 is the longitude of the second station in radian.

Positive latitude represents northern hemisphere while negative latitude represents southern hemisphere.

Positive longitude represents eastern hemisphere while negative longitude represents western hemisphere.

Input

The first line of input contains an integer $t (1 \le t \le 100)$ - the number of test cases.

The first line of each test case contain an integer $n (3 \le n \le 200)$ - the number of stations.

The subsequent line of each test cases contains the integer input of the geographical location of n stations in latitude a and longitude b where $(-90 \le a \le 90)$ and $(-180 \le b \le 180)$.

Output

For every test case, print the **minimum total maintenance cost** in integer after applying floor function.

Example

standard input	standard output
2 3 -59 -130 90 -149 -4 44 4 -83 -71 59 122 64 153 -5 50	23442 21043

Problem H. Red Notice

Input file:	`standard input`
Output file:	`standard output`
Time limit:	4 seconds
Memory limit:	256 megabytes

Nolan Booth is the most skilful thief in the world, renowned for his exceptional stealing skill. One day, he break through the securities of a museum successfully and there are n rooms in the museum, each room have k piles of treasures.

Since the security guards are having a break and will come back in h hours, he want to steal all the treasures undetected. Therefore he can decide his treasure stealing speed per hour, which is m. Each hour, he will enter a room and steal m treasures from the room. If the room has less than m treasures, he will steal all of them and will not steal anymore treasures during this hour.

He likes to steal slowly but still wants to steal all the treasure before the security guards return. What is the **minimum** treasure stealing speed per hour such that he can steal all the treasures?

Input

The first line of input contain an integer t ($1 \leq t \leq 100$) - the number of test cases.

The first line of each test case contains an integer n ($1 \leq n \leq 10^5$) - the number of rooms in the museum.

The second line of each test case contains an integer h ($n \leq h \leq 10^9$) - the number of hours for the guards to come back from break.

The third line of each test case contains n integers, $k_1, k_2, ...k_n$ where ($1 \leq k \leq 10^9$)

Output

For every test cases, print the **minimum** treasure stealing speed per hour.

Example

standard input	standard output
2 7 8 75 52 68 57 61 32 67 3 9 53 9 21	68 11

Problem I. Security System

Input file:	`standard input`
Output file:	`standard output`
Time limit:	1 second
Memory limit:	256 megabytes

Meow is being tasked with creating a brand new password security system. It is mainly used to overcome weak password storage solution in today's market. His system works in this way:

1. There are n memory storage areas.
2. Each storage area can hold any number of signals, including no signal – **NULL**
3. There are 2 different signals – "0"and "1"
4. There exists a "0" signal and b "1" signals.
5. Meow will place these signals into the memory storage areas. Meow can choose to place **some or all or none** of those "0"and "1"signals.
6. The order of signals inside a specific memory storage area does not matter, but order matters across different memory storage areas.
7. A new password is generated by having a unique combination of storing the signals.

Meow would like to know how many different password can the system have?

Input

The input contains 3 integers n, a, b $(0 \le a, b \le 500), (1 \le n + a, n + b \le 500)$ – the number of memory storage, the number of "0"signal available, and the number of "1"signal available.

Output

Print a single integer – the number of passwords the system can have modulo $10^9 + 7$ (1000000007).

Example

standard input	standard output
`2 1 1`	`9`

Note

There are two memory storage area, and Meow has only one "0"signal and one "1"signal to work with. The 9 possible storage options are listed below:

	Memory 1	Memory 2
1	NULL	NULL
2	0	NULL
3	1	NULL
4	NULL	0
5	NULL	1
6	01	NULL
7	NULL	01
8	1	0
9	0	1

Note that the order inside a memory storage area does not matter, for the 6^{th} and 7^{th} storage option, 01 and 10 are considered similar.

Problem J. UM Librarian

Input file:	`standard input`
Output file:	`standard output`
Time limit:	2 seconds
Memory limit:	256 megabytes

You want to become junior librarian in University of Malaya Library. A senior librarian wants to give you a test before you can be recruited as a junior librarian by giving a books list of size n which contains $d_1, d_2, ...d_n$ books ID, the order of the books should be in. You are also given a book rack of size m containing $b_1, b_2, ..b_m$ books.

Each book have an ID and you are required to arrange the books in a book rack according to the books ID in the list. The books with book ID which is at the lower index position in the book list will have a higher priority to be arranged in front of the book rack.

When arranging the books, you are only allowed to swap any two books **adjacently**. Since you are determined to become a junior librarian and want to impress the senior librarian. How can you arrange the books in **minimum swap**?

Input

The first line of input contains a single integer t $(1 \leq t \leq 100)$ - number of test cases.

The first line of every test case contain an integer n $(3 \leq n \leq 10000)$ - size of book list.

The second line of every test case contains n integers, representing the book ID in the book list where $d_1, d_2, ...d_n$ $(1 \leq d_i \leq n)$.

The third line of every test case contains an integer m where $(m = 2 \times n)$ - The size of the book rack.

The fourth line of every test case contains m integers, representing the books in the book rack where $b_1, b_2, ...b_m$ $(1 \leq b_i \leq n)$. There may have **duplicated** books ID in the rack, and some books ID in the list may **not exist** in the rack.

Only one valid answer exist.

Output

For every test case, print the **minimum number** of adjacent swaps to arrange the books according to the book list.

Example

standard input	standard output
3 3 1 2 3 6 1 3 1 3 2 3 3 3 1 2 6 2 2 2 2 3 3 3 3 1 2 6 1 1 3 3 2 2	3 8 4

Note

For the 1^{st} test case, after sorting, the book rack will be 1 1 2 3 3 3

This is one of the sorting method :

- 1 **1** **3** 3 2 3
- 1 1 3 **2** **3** 3
- 1 1 **2** **3** 3 3

For the 2^{nd} test case, after sorting, the book rack will be 3 3 2 2 2 2

For the 3^{rd} test case, after sorting, the book rack will be 3 3 1 1 2 2