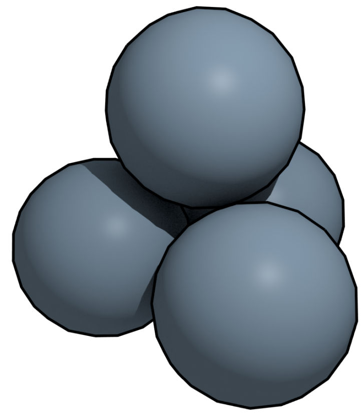
Problem A
Cannonball Pyramids
Time Limit: 2 seconds



Fort Calgary seems to have an excess of cannonballs left over from its war with the whiskey traders of the 19th century, and has tasked Peter with stacking them up for storage. As a computer scientist, Peter thought he'd develop an Automatic Cannonball Pyramid Creator (ACPC) to make the task easier.

Cannonballs are all perfectly spherical and exactly the same size. Peter will always stack the cannonballs in a triangular pyramid for storage (see diagram). Floor space is at a premium, so Peter would like to use the smallest triangular base (or part thereof) possible for the stack. Once the base is established, it'd be nice to stack the cannonballs in as few levels as possible so that we wouldn't have to reach so high to retrieve them.

Can you write a program to determine the smallest height of the pyramid, given the number of cannonballs to stack, to help Peter develop the ACPC?

Input

The input file starts with an integer $T (1 \leq T \leq 1000)$, the number of test cases. Each test case consists of a single integer $N (1 \leq N \leq 10^{15})$, the number of cannonballs.

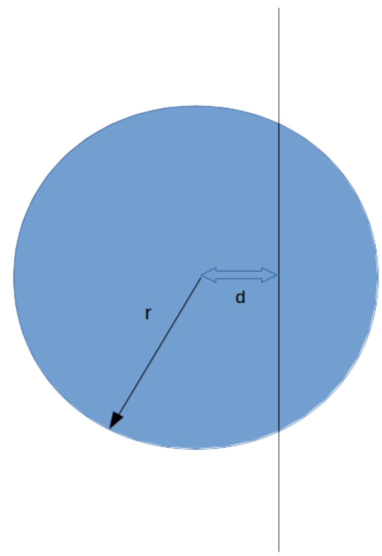
Output

For each test case, output the minimum number of levels that you need in order to stack N cannonballs using the smallest base possible (or a portion of it, see examples).

Sample Input	Sample Output
4 2 4 5 10	1 2 1 3

This page is intentionally left blank.

Problem B
Lenses
Time Limit: 2 seconds



Astronomy Club at Physics Center (ACPC) has hired you for a rather difficult task. They want you to make some lenses for their telescopes and observation tools. They would give you n glass spheres where the ith sphere has radius r_i. You would then place a virtual plane at distance d_i from the center of the sphere and cut on that plane using laser and give the smaller part to ACPC technicians. Do you care about the remaining of the sphere? Of course not! But you want to know how much glass you have wasted. So, find out!

Input

First line of input contains a single integer $0 < n \leq 1000$, the number of spheres. Lines $2..n+1$ describe the spheres. Specifically, line $i+1$ contains integers $1 \leq r_i \leq 50$ and $0 < d_i < r_i$, the radius of sphere i and its cutting distance respectively.

Output

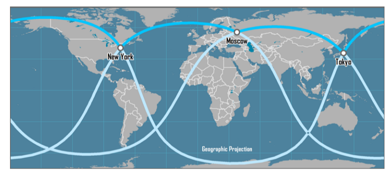
Print n lines, each including a single number for each sphere indicating the volume of wasted glass to 2 decimal points.

Sample Input	Sample Output
2 5 1 10 2	339.29 2714.34

This page is intentionally left blank.

Problem C
The Shortest Path
Time Limit: 2 seconds

Given a string S, your task is to find an occurence of the string T="path" as a subsequence in S, such that T spans the least number of characters in S.



Input

The input file starts with an integer $N (1 \leq N \leq 100)$, the number of test cases. Each case contains a string S on a separate line. $(4 \leq |S| \leq 500, s_i \in \{\text{'p','a','t','h'}\})$

Output

For each test case output the minimum number of characters in S that the string T is 'spanning' as a subsequence, or -1 if T is not a subsequence of S.

Sample Input	Sample Output
3 path htap hptatptahaha	4 -1 8

This page is intentionally left blank.

Problem D
Student Clubs
Time Limit: 2 seconds

At the University of Calgary, it is time to elect student officials and the board that will regulate and oversee the functioning of student clubs. Some clubs are smaller, some larger, so the consensus was that the members of the board will be chosen from all the eligible club members uniformly at random.

Can this system really be fair to the smaller clubs?

Input

On the first line of input you are given an integer $T (1 \leq T \leq 100)$, the number of test cases. Each test case will consist of two lines. On the first line you will be given integers C and B, the number of clubs and the size of the board, respectively $(2 \leq C \leq 18, B)$. The second line will contain C integers, the number of eligible members from each club c_i $(1 \leq c_i \leq 18, B \leq \sum_{i=1}^{C} c_i \leq 60)$.

Output

For each test case, output the probability that the board will have at least one representative from each student club, rounded to 6 decimal places.

Sample Input

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

Sample Output

```
0.666667
0.666667
0.400000
0.300000
```

This page is intentionally left blank.

Problem E
Aggregating Points
Time Limit: 6 seconds

One fundamental problem in computing science is clustering points in a cheap way. This is usually used to group "similar" objects together. Here, you will deal with a simple version.

You are given n points P scattered throughout the number line. You suspect that these n points were obtained by multiple "noisy" observations of just k points. Thus, you want to find k points C on the line such that the maximum distance between some point of P and its nearest point on C is as small as possible. Actually, you only care about the minimum such distance and not the actual locations of the k points.

Input

The first line contains a single integer $T \leq 10$ indicating the number of test cases. The first line of each test case consists of two integers n and k satisfying $1 \leq k \leq n \leq 10^5$. The final line of each test case consists of n integers $x_1, \ldots, x_n$ satisfying $1 \leq x_1 \leq x_2 \leq \ldots \leq x_n \leq 10^9$. These numbers are the n points P mentioned above. Consecutive numbers on this line will be separated by a single space.

Output

For each test case, output a single floating point d number with exactly one decimal place. This number d should be the smallest number such that there is some set C of k numbers where every x_i in the input is at most distance d from some point in C.

Sample Input	Sample Output
3 3 2 1 2 3 3 3 1 2 3 6 2 1 5 5 10 20 30	0.5 0.0 5.0

This page is intentionally left blank.

Problem F
Coffee Beans
Time Limit: 20 seconds

Having wasted enough time with boring assignments and useless classes, you've decided to drop out of school and follow your true passion, coffee. You're going to open your own coffee shop and roasting company, the Alberta Coffee Makers. With a shop leased and roasting equipment acquired, all that's left is some simple math to prepare for opening day!

You've done some market research and you know that you can sell a cup of coffee for $\$2.50$ and that you will sell N cups each day. One pound of roasted coffee beans makes 16 cups of coffee, so you will need to buy and roast $7 * N/16$ pounds of beans each week.

Your bean supplier has M different kinds of beans for sale. Though he has enough of each kind of bean to fulfill your weekly needs, he'd prefer you buy more than one type so that there's enough for all the other roasters he supplies. For this reason, he will charge you $c_i * w + d_i * w * w$ for w pounds of bean i. He'll let you buy as many different kinds of beans as you like, and you may buy in fractions of a pound as well. He'll even deliver your order for free each week!

You've decided to roast all your beans on Sunday before the work week begins. Each kind of bean needs to be roasted in a separate batch and at a different temperature. For example, lighter coffees take less time and energy to roast than darker coffees. It will cost you $r_i * w$ to roast w pounds of bean i.

Your aim, obviously, is to maximize your weekly profit. How much of each bean should you buy to accomplish that goal?

Input

The input file starts with an integer $T (1 \leq T \leq 1000)$, the number of test cases. Each test case starts with two integers N and M on a line. M lines follow with three floating point numbers each c_i, d_i and r_i (each will have at most two digits after the decimal point).

$0 < N <= 3500$

$0 < M <= 10000$

$0.10 <= c_i, d_i, r_i <= 15.00$

Output

For each test case, output the maximum profit on a separate line, rounded to the nearest whole cent. Make sure you do not 'buy' negative amounts of coffee (although profit itself may be negative).

Sample Input	Sample Output
2 16 2 10.00 0.10 1.00 11.50 0.05 1.25 16 2 10.00 0.10 1.00 11.50 0.05 0.50	198.10 198.37

Problem G
Square Grids
Time Limit: 2 seconds

Given a square grid that has some (but not all) of its tiles filled, what is the largest square grid that can be placed on top of the original grid in such way that it does not cover any of the filled tiles?

Input

The input file starts with an integer $T (1 \leq T \leq 100)$, the number of test cases. Each test case starts with a line containing an integer $N (2 \leq N \leq 2000)$, the size of the original square grid, followed by N lines containing N characters from {'E','F'}, 'E' indicating an empty tile and 'F' the filled one.

Output

For each test case, output the size of the square that can be placed in the original grid without covering any previously filled tiles.

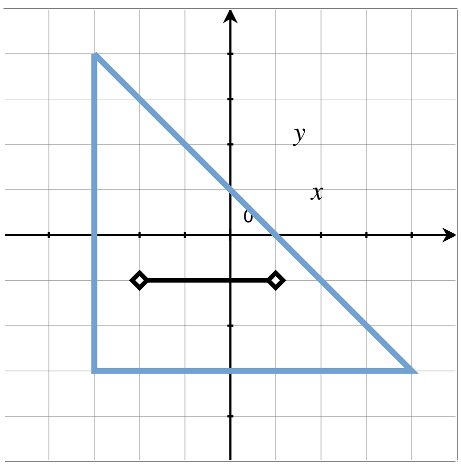
Sample Input	Sample Output
2 2 EE EE 3 FEE EEE EEE	2 2

This page is intentionally left blank.

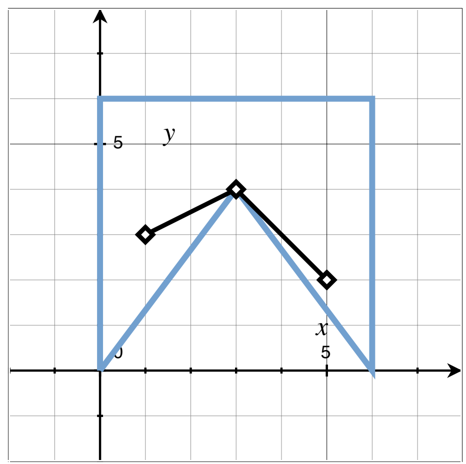
Problem H
Dad to the Rescue!
Time Limit: 5 seconds

"Waaaa! <cough> <cough>" cried little Henry. He just caught a rare condition called Ate a Cherry Pit and Choked (ACPC)! Dad now needs to run over to Henry as fast as he can to rescue him using the Heimlich manoeuvre.

Henry lives in a single storey house, but the house is very large with many nooks and crannies (though, oddly enough, no furniture to get in the way). We can describe the floor plan to you by tracing all its inside walls, which will always form a simple polygon. If you know exactly where Henry and Dad are within their house, can you help Dad find the shortest path to his son?



Test Case #1



Test Case #2

Input

The input file starts with an integer $T (1 \le T \le 100)$, the number of test cases. Each test case starts with an integer $N (1 \le N \le 300)$ on a line, followed by $N + 2$ lines. First N lines contain pairs of integer coordinates, describing the walls, followed by the locations of Henry and his Dad respectively on the next two lines. All coordinates have magnitude less than 1000.

Each wall of the house is described by its endpoints, (x_i, y_i) and (x_{i+1}, y_{i+1}), with all walls consecutive in a counter-clockwise winding. This means that if you're standing at the first endpoint and looking at the second, the space on your left side is the interior of the house. The endpoints (x_N, y_N) and (x_1, y_1) form the final wall closing the house. As indicated, the walls do not intersect, and Henry and Dad are both *strictly* inside the house, meaning their initial positions are not in contact with any walls or corners of the room. However, Dad may run snug against walls and corners on his path to Henry.

Output

For each test case, output the length of the shortest path between Henry and his Dad completely contained within their house, rounded to 6 decimal places.

Sample Input	Sample Output
2 3 -3 4 -3 -3 4 -3 -2 -1 1 -1 5 0 0 3 4 6 0 6 6 0 6 5 2 1 3	3.000000 5.064495

Problem I
Election Campaign
Time Limit: 10 seconds

As you know, it is time for another exciting election! This year, the campaign signage is even more absurd than usual. Perhaps the most ridiculous aspect is how posters have been haphazardly placed on public walls. In particular, there is one very long wall that many posters have been placed on. People have rudely slapped up posters promoting their favourite candidate and many of these cover parts of previously-placed posters.

Each poster's height is exactly as tall as the wall, but their horizontal spans can vary. In particular, a poster i is given by the x-coordinates of its left and right edges, say ℓ_i and r_i. We then say that poster i covers all points x such that $\ell_i < x < r_i$.

Now consider posters i and j with edges at ℓ_i, r_i and ℓ_j, r_j respectively. Say that poster i *partially obscures* poster j if poster i was placed after poster j and if there is some point x covered by both i and j but not covered by any other poster placed before i and after j.

Note that this is not the same as simply saying poster i covers some point that is also covered by poster j. We don't want to blame i for covering part of j if this part was already covered by other posters placed before i.

You want to see how rudely the posters are placed. For each poster i, you should find the list of all other posters j such that poster i partially obscures poster j.

Input

The first line of input contains a single integer $T \leq 10$ indicating the number of test cases. A test case begins with a single integer $1 \leq n \leq 10^5$ indicating the number of posters in the input. Then n lines follow, each describing a poster. The ith such line describes the ith poster that was placed on the wall and this is given by two coordinates ℓ_r, r_i satisfying $1 \leq \ell_1 < r_i \leq 10^9$.

Output

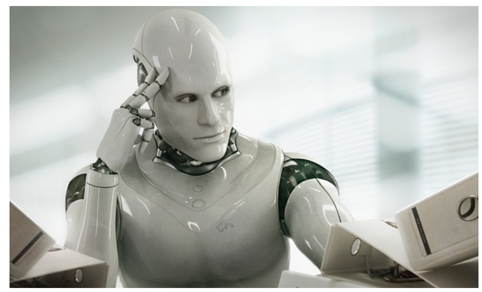
The output for each test case consists of $n + 1$ lines. The first line is simply `Case #:` where # indicates which test case is being processed. The remaining n lines describe how posters are directly obscured.

The ith such line begins with a number s_i indicating how many posters are directly obscured by poster i. On the same line, these s_i posters are given in increasing order. A single space should appear between every pair of consecutive numbers on this line.

Sample Input	**Sample Output**
3 3 0 5 5 10 2 6 5 0 2 1 3 2 4 3 5 4 6 3 1 2 1 2 1 2	Case 1: 0 0 2 1 2 Case 2: 0 1 1 1 2 1 3 1 4 Case 3: 0 1 1 1 2

Problem J
Artificial Intelligence
Time Limit: 2 seconds

Your pet robot project is coming along nicely. It is time to teach Abraham Charles Maximillian (Abe, for short) how natural numbers are ordered. He somehow figured the way natural numbers were created and for some reason he is insisting that zero be included, but you ignore him.



The way you try to train Abe is - show him two cards containing two different natural numbers then place the third card containing either ‘greater than’ (>) or ‘less than’ (<) sign.

Your briliant and, more importantly, novel idea is that, eventually, Abe will pick up what kind of relationship between two natural numbers these symbols represent and deduce a lot of other useful things and develop new skills from there, including the ability to estimate the airspeed velocity of an unladen swallow, regardless of its continent of origin.

It is time to get to work:

1. Show him two numbers and see what symbol (if any) Abe puts between them.
2. Repeat 1. until Abe is definitely ready.
3. Fix that zero nonsense.
4. Sell your robot to PetroFeed.
5. *Cha-ching.*

We will focus on the step 1. at this point.

Input

The input file starts with an integer $T(1 \leq T \leq 100)$, the number of test cases. Each test case consists of a line containing two integers, A and B ($1 \leq A, B \leq 20, A \neq B$) and an operator (either ‘<’ or ‘>’).

Output

For each test case, if Abe correctly marked the relationship between A and B output “Ready”, otherwise output “Reboot” (without quotes). Output for each test case should be on a separate line.

Sample Input	Sample Output

`2`
`1 2 <`
`3 4 >`
	`Ready`
	`Reboot`

This page is intentionally left blank.