The Weary Butcher

In the city of Bertaku, butchery is considered to be a very tough job, and thus there is only one butchery in the city. Customers have way strange requests, and the lonely butcher has only one heavy big dull cleaver – a butcher-knife.

Each stroke of the cleaver cuts the meat in a straight line in the same direction of the stroke. The knife is very large – in fact larger than any piece of meat – therefore each stroke splits the meat in two pieces.

Each customer draws a convex polygon on a rectangle piece of meat, and asks the butcher to cut the polygon out of it. So the butcher has to make some strokes and throw some pieces of meat away, and deliver a piece of meat with the desired shape to the customer.

In each stroke, the butcher has to use 100 units of force per each centimeter of meat his knife is cutting. Considering butcher's weariness, he wants to minimize his force in delivering each request.

You are given a customer's requested polygon and the size of a rectangular meat. Calculate the minimum units of force the butcher has to use to deliver the request.

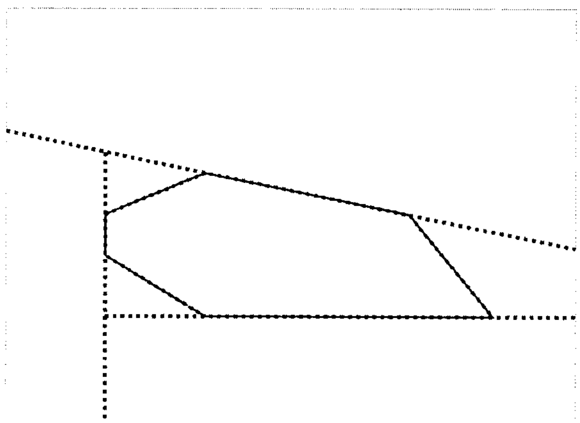

Figure 1: Illustration of sample test

Input

The first line of input contains a single integer $1 \leq T \leq 20$, the number of test-cases.

It is followed by T test-cases, each starting with a line containing three space-separated integers W, H, N $(0 \le W, H \le 10^4; 3 \le N \le 150)$, where W and H are width and height of the rectangular meat and N is number of vertices of the requested polygon.

The next N lines contain two integers each, representing coordinates of the polygon's vertices in clockwise order. The i-th line will contain two integers x_i and y_i $(0 < x_i < W; 0 < y_i < H)$. The lower-left corner of the rectangle is located at $(0, 0)$ and the upper-right one at (W, H).

Output

For each test-case, print one line containing the minimum amount of force (with exactly two digits after decimal point) the butcher should apply to cut out the shape.

Sample Input/Output

Input	Output
1 140 100 6 50 60 100 50 120 25 50 25 25 40 25 50	41086.88

Permutation

Arash and Babak are two mathematicians playing an interesting game. Arash chooses a permutation of numbers 1 to N in his mind, and Babak tries to guess it. In each turn, Babak guesses a permutation and Arash tells him the number of elements having the same place in both permutations. The game ends when Babak finds the permutation Arash had chosen.

Surprisingly, we have completely modeled Arash as a computer program, and we've made it available to you. You should implement a model for Babak as a computer program as well. In each turn, your program should print (in standard output) its guessed permutation as N space-separated numbers in one line. Afterwards, it should read a single integer (number of correctly-placed elements of the guessed permutation) from standard input.

Our implementation of Arash will execute your program and communicate with it automatically. You can learn more about Arash by running "`arash --help`"

Input

A secret key which `arash` uses to evaluate your program.

Output

If your program guesses the permutation correctly, our provided application (`arash`) will generate a secret output which you should submit.

Sample Input/Output

Intentionally left blank!

bayAndroid

Maryam recently bought a bayAndroid mobile phone, and was so excited about its fabulous "idleLock" feature. bayAndroid falls to a locked mode after a period of idleness, in which a $4 \times m$ grid is displayed on the screen, and the user should draw a preset "pattern" to unlock it. Of-course, not all "drawings" on the grid are patterns! After a bit of trial-and-error, Maryam found out that a valid unlock pattern:

- Must start at the lower-left point of the grid, and end at the upper-right point.
- Every two consecutive points of it must be adjacent[1] in the grid.
- Must not cross or touch itself.

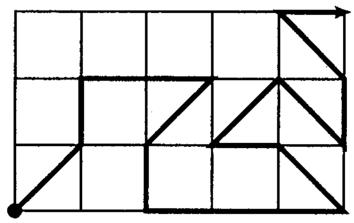

Figure 2: A valid pattern

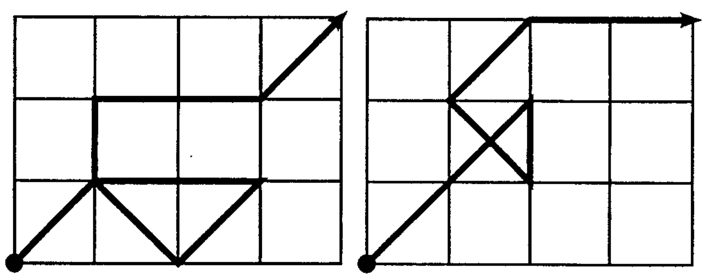

Figure 3: Two invalid patterns: the left one touches itself and the right one crosses itself.

Not much after his valuable "discovery" was she thinking about the number of valid unlock patterns. Unfortunately, Maryam is too busy doing her homeworks right now, so you were chosen – by the problem-set organizer of-course – for solving this critical task.

[1]Each interior point of the grid has 8 adjacent points.

Input

The first line of input contains the number of tests $1 \leq T \leq 10$.

Each of the following T lines contains the only number $2 \leq m \leq 10^{15}$ – width of the grid.

Output

For each test-case, output the number of different valid patterns modulo $10^9 + 7$ on its own line.

Sample Input/Output

Input	Output
1 2	89

Inferno of Impact

Consider a non-weighted graph G with N vertices and M bi-directional edges. Note that multiple edges (edges with same endpoints) are allowed.

We'll build a simple weighted graph F from G with the following rules:

- $V_F = V_G$
- For each pair of vertices in F, (u, v), there is an edge between u and v with its weight equal to the minimum-cut between corresponding vertices in G.

Given G and two constant integers K and W, calculate number of vertex-induced subgraphs of F with exactly K vertices, such that sum of its edges equals W.

Input

The first line of input contains a single integer T, the number of test-cases.

It is followed by $1 \leq T \leq 10$ test-cases, each starting with a line containing four space-separated integers N, M, K, W ($2 \leq N \leq 60; 0 \leq M \leq \binom{N}{2}; 2 \leq K \leq N$).

Each of the next M lines contains two space-separated integers u_i and v_i indicating an edge between vertices u_i and v_i in G. ($1 \leq u_i, v_i \leq N$)

Output

For each test-case, in one line print a single integer the task asks for.

Sample Input/Output

Input	Output
2 4 6 1 2 5 2 2 3 3 4 2 3 5 6 5 5	10 14

The Museum

A robber wants to rob a museum in a night. He places a lamp on the floor of the museum room and robs every illuminated thing!

The room has many columns in shape of circles and convex polygons that are dark objects (i.e. light does not pass them). Nothing in this room reflects the light. There are no overlaps between any two columns and the columns do not touch the museum's walls.

Calculate the area of the floor that the lamp illuminates.

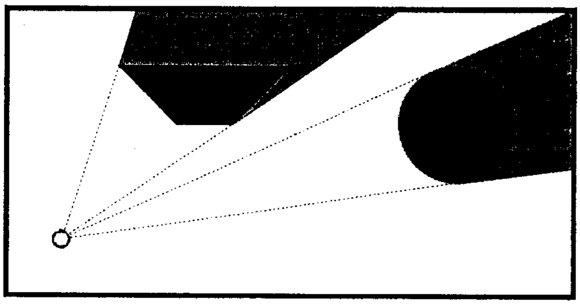

Figure 4: Illuminated area is colored in white, columns are black and shadowed area is in gray

Input

The first line contains a single integer $1 \leq T \leq 10$, the number of test-cases.

It is followed by T test-cases, each starting with a line containing five space-separated integers W, L, K, X, Y where $3 \leq W, L \leq 10^9$ are width and length of the museum (i.e. corners of the room are located at $(0,0)$ and (W, L)), and $0 \leq K \leq 100$ is number of columns and $0 < X < W$ and $0 < Y < L$ are coordinates of the lamp.

Then next K lines will contain properties of columns.

Properties of a polygonal column starts with a line containing a character `'P'` followed by an integer $3 \leq n \leq 10$, the number of vertices of the column, followed by n lines, each containing two space-separated integers $0 < x_i < W$ and $0 < y_i < L$, representing coordinates of i-th vertex. Vertices of the columns are in clock-wise order. No three vertices of a polygon are co-linear.

Properties of a circular column starts with a line containing a character `'C'` followed by three space-separated integers $0 < x < W$, $0 < y < L$ and r, describing coordinates of center of the i-th circular column and its radius.

Output

For each test-case, print a line containing the percentage of the area of the museum room the lamp illuminates, with exactly 2-digits after decimal point.

Sample Input/Output

Input	Output
1 10 5 2 1 1 P 4 2 4 5 4 4 3 3 3 C 8 3 1	74.55

Sample test is illustrated above.

NAM

The Non-Aligned Movement (NAM) is a group of politically independent countries, having about 120 members. Last year (2012) the summit was held in Tehran, Iran. We want to see how you could help the Iranians to organize the summit as a programmer. The organizers wanted to invite as many presidents as possible to the summit session.

Iran has N cities $\{C_1, \ldots, C_N\}$, C_N being the capital, Tehran, and M one-way roads between these cities. Suppose there are P countries in the NAM which their president may be invited to the summit. The i-th country's president – if invited – could arrive to Iran with a flight that lands on the city C_{A_i} at time Z_i. He then had to travel to Tehran by traveling the roads. It takes one hour to travel each road. For security reasons, no two presidents may be on the same road in any time interval, but it is okay to be in the same city. Also they might stay in a city as much as they'd like to. The session would be held at time S.

The organizers didn't want any president to be late (strictly speaking, no invited president may arrive Tehran after time S), so they have asked you to compute maximum number of presidents that could be invited to the session.

The organizers, of-course, would be happy to start the session sooner. So you have to compute the soonest time that the session could start with maximum attendants, as well. (i.e. Their first priority is the number of attendants, and the second is to start the summit as soon as possible.)

Input

The first line of input contains a single integer $1 \leq T \leq 10$, the number of test-cases.

It is followed by T test-cases, each starting with a line containing two space-separated integers N, M ($1 \leq N \leq 100; 0 \leq M \leq 500$). The next M lines each contains two integers u_j and v_j, indicating a one-way road between two cities u_j and v_j. The test case continues with a line containing two space-separated integers, $0 \leq S \leq 500$ and $0 \leq P \leq 120$ and the next P lines each contains two space-separated integers, A_i and Z_i.

Output

For each test-case, print one line containing two space-separated numbers. The first one being the number of presidents that can be invited, and the second being the soonest time that the summit may start.

Sample Input/Output

Input	Output
1 4 6 1 2 2 1 1 3 3 4 4 2 2 3 10 5 1 0 1 0 2 0 2 0 3 4	5 6

The Sorter Machine

The sorter machine was a machine built around 1950. It was used to sort a permutation of numbers $< 1, 2, \cdots, N >$ given as input in ascending order using a preset program. The program consisted of M pair of integer between 1 and N. The sorter machine read the program from beginning to the end, and for the i-th pair represented by (a_i, b_i), compared a_i-th and b_i-th numbers in the input permutation, and if these two numbers were not in order, swapped them.

The sorter machine was later found by a group of geologists at February 2013, but since they were not familiar with Computer Science, they called programmers from all around the world to write a sorter machine program that correctly sorts all permutations.

You are given a few of these programs. Determine whether they work correctly or not.

Input

The first line of input contains a single integer $1 \leq T \leq 30$, the number of programs.

It is followed by T programs, each starting with a line containing two space-separated integers N, M ($1 \leq N, M \leq 60$), where N is the size of permutation given to the sorter machine and M is the number of pairs in the program. The next M lines contain two integers each, a_i and b_i, ($1 \leq a_i, b_i \leq N$) – explained in the problem statement.

Output

For each program, if the program is correct, print "YES" in one line. Otherwise, print "NO", followed by lexicographically-smallest permutation of numbers $< 1, 2, \cdots, n >$ that the program can not sort correctly.

Sample Input/Output

Input	Output
2 3 2 1 2 2 3 3 3 1 2 1 3 2 3	NO 2 3 1 YES