# Speedway

Given a vehicle and an oval circuit surface you are requested to compute a route around the track for a vehicle, making it as fast as possible.

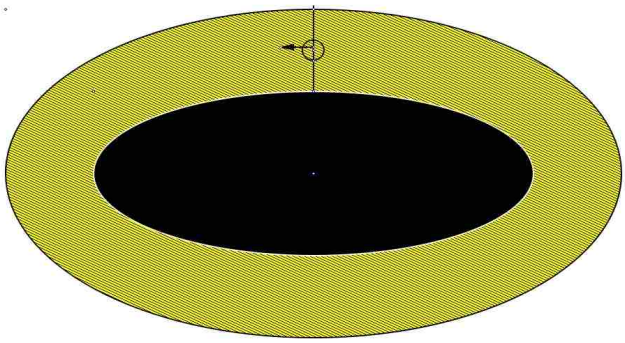
The track is bounded by two ellipses: the outer one and the inner one. You must keep the vehicle inside the outer ellipse and outside the inner ellipse. The centers of both ellipses are placed in the same point (0, 0). The lengths of the semi-axes are:

- *hIW* - the length of the semi-axis parallel to the x-axis of the inner ellipse
- *hIH* - the length of the semi-axis parallel to the y-axis of the inner ellipse
- *hOW* - the length of the semi-axis parallel to the x-axis of the outer ellipse
- *hOH* - the length of the semi-axis parallel to the y-axis of the outer ellipse

Your vehicle is very simple and at every moment it is able to perform only one action among the two:

- to **turn_left** - the vehicle turns left by the angle *a* and slows down (decelerates).
- to **speed_up** - the vehicle accelerates keeping the previous direction.

Acceleration and deceleration rates are parameters of the vehicle: acc, dec.



## Simulation

The simulation proceeds in steps. At the beginning (step 0) the vehicle is placed at point (0, (*hIH* + *hOH*)/2) with initial velocity = [-1, 0]. The coordinates and velocity of your vehicle in the next steps depend on the previous coordinates, its velocity, and the command it is following.

The vehicle coordinates in the (*n*+1)-th step are simply coordinates in the *n*-th step plus *velocity* (where velocity is a vector, thus addition is performed separately for the horizontal and vertical coordinates).

If the *n*-th action is **speed_up**, than the vehicle's velocity in the (*n*+1)-th step is: (|*velocity*| + *acc*) / |*velocity*|) *velocity*.

If the *n*-th action is **turn**, than the vehicle's velocity in the (*n*+1)-th step is: (|*velocity*| - *dec*) / |*velocity*|) *velocity R*, where *R* is the two-dimensional rotation matrix by *a*degrees.

You control your vehicle by two commands: s for **speed_up** and t for **turn_left**. We will assume that the vehicle never stops and if you command it to stop (or something close to stop, that is you

give the command t while the vehicle's speed is at most *dec*+0.0001), the vehicle will accelerate anyway.

Simulate the process using the single-precision floating-point variables (e.g. float in C, C++, Java, ...).

## Input

The first line contains the number of test cases *t*. Each of the following *t* lines contains: 4 integers $10 < hIW, hIH, hOW, hOH < 1000$, $hIW<hOW$ and $hIH<hOH$ (describing the arena), and 3 positive numbers *acc*, *dec*, *a* describing the vehicle.

## Output

For each test case output the number *k* of steps which could have led from the initial vehicle position to the completion of one lap of the track and in the next line, the description of those steps (please consult the example below) or one word NO if you do not want to solve this particular test case.

## Scoring

The score awarded to your program for a given test case is computed as 1000-*k*.

The score awarded for a given test set is computed as a maximum of 0 and the sum of scores for individual test cases. The overall score of the program is the sum of scores obtained for correctly solved tests.

The number of points given in the ranking is scaled so that it is equal to 10 for the registered contestant whose solution has the highest score, and proportionally less for all solutions with lower scores.

## Example

**Input:**
2
300 150 450 300 0.3 0.7 15.0
300 150 320 170 0.3 0.3 5.0

**Output:**
142
sssssssssssssssssssssssstsssssstsssssstsssssstsssssssssstsst
ssstsstssstststsssstsssstsssssssssssssssssstsssssststsstssts
tststsssstsststsssssstssssss [the whole string in one line]
NO

**Scoring:**
test1 1000-142=**858** points
test2 skipped
set score: **858** points.

The trace for the above route is available here.

## Input data

| n | t | k | l |
|---|---|---|---|
| 1 | 1 | >40 | 2s |
| 2 | 6 | >70 | 2s |
| 3 | 10 | >100 | 2s |
| 4 | 10 | >350 | 2s |
| 5 | 10 | ?? | 5s |
| 6 | 10 | ?? | 5s |
| 7 | 10 | ?? | 5s |

(some cases might be unsolvable)

n - test set number
t - the amount of test cases in the test set
k - the expected number of steps to be taken
l - time limit

## Please note

- Till the last week of the series, all submitted codes will be visible to all users and tested on temporary data sets only.
- For the last week of the series, submissions will be visible to the submitting contestant, only, and tested on the full set of test cases.
- After the deadline **datasets will be slightly modified** and all solutions will be rejudged.
- Please do not submit more than 100 times to this problem (all submissions starting with the 101st will be ignored).