Codeforces Beta Round #23

A. You're Given a String...

time limit per test: 2 seconds
memory limit per test: 256 megabytes
input: standard input
output: standard output

You're given a string of lower-case Latin letters. Your task is to find the length of its longest substring that can be met in the string at least twice. These occurrences can overlap (see sample test 2).

Input
The first input line contains the string. It's guaranteed, that the string is non-empty, consists of lower-case Latin letters, and its length doesn't exceed 100.

Output
Output one number — length of the longest substring that can be met in the string at least twice.

Examples

input
abcd
output
0

input
ababa
output
3

input
zzz
output
2

B. Party

time limit per test: 2 seconds
memory limit per test: 256 megabytes
input: standard input
output: standard output

n people came to a party. Then those, who had no friends among people at the party, left. Then those, who had exactly 1 friend among those who stayed, left as well. Then those, who had exactly $2, 3, \ldots, n - 1$ friends among those who stayed by the moment of their leaving, did the same.

What is the maximum amount of people that could stay at the party in the end?

Input

The first input line contains one number t — amount of tests ($1 \le t \le 10^5$). Each of the following t lines contains one integer number n ($1 \le n \le 10^5$).

Output

For each test output in a separate line one number — the maximum amount of people that could stay in the end.

Examples

input
1 3
output
1

C. Oranges and Apples

time limit per test: 2 seconds
memory limit per test: 256 megabytes
input: standard input
output: standard output

In $2N - 1$ boxes there are apples and oranges. Your task is to choose N boxes so, that they will contain not less than half of all the apples and not less than half of all the oranges.

Input

The first input line contains one number T — amount of tests. The description of each test starts with a natural number N — amount of boxes. Each of the following $2N - 1$ lines contains numbers a_i and o_i — amount of apples and oranges in the i-th box ($0 \le a_i, o_i \le 10^9$). The sum of N in all the tests in the input doesn't exceed 10^5. All the input numbers are integer.

Output

For each test output two lines. In the first line output YES, if it's possible to choose N boxes, or NO otherwise. If the answer is positive output in the second line N numbers — indexes of the chosen boxes. Boxes are numbered from 1 in the input order. Otherwise leave the second line empty. Separate the numbers with one space.

Examples

input
2 2 10 15 5 7 20 18 1 0 0
output
YES 1 3 YES 1

D. Tetragon

time limit per test: 3 seconds
memory limit per test: 256 megabytes
input: standard input
output: standard output

You're given the centers of three equal sides of a strictly convex tetragon. Your task is to restore the initial tetragon.

Input

The first input line contains one number T — amount of tests ($1 \le T \le 5 \cdot 10^4$). Each of the following T lines contains numbers x_1, y_1, x_2, y_2, x_3, y_3 — coordinates of different points that are the centers of three equal sides (non-negative integer numbers, not exceeding 10).

Output

For each test output two lines. If the required tetragon exists, output in the first line YES, in the second line — four pairs of numbers — coordinates of the polygon's vertices in clockwise or counter-clockwise order. Don't forget, please, that the tetragon should be strictly convex, i.e. no 3 of its points lie on one line. Output numbers with 9 characters after a decimal point.

If the required tetragon doen't exist, output NO in the first line, and leave the second line empty.

Examples

input
3 1 1 2 2 3 3 0 1 1 0 2 2 9 3 7 9 9 8

output
NO YES 3.5 1.5 0.5 2.5 -0.5 -0.5 2.5 0.5 NO

E. Tree

time limit per test: 2 seconds
memory limit per test: 256 megabytes
input: standard input
output: standard output

Recently Bob invented a new game with a tree (we should remind you, that a tree is a connected graph without cycles): he deletes any (possibly, zero) amount of edges of the tree, and counts the product of sizes of the connected components left after the deletion. Your task is to find out the maximum number that Bob can get in his new game for a given tree.

Input

The first input line contains integer number n ($1 \le n \le 700$) — amount of vertices in the tree. The following $n - 1$ lines contain the description of the edges. Each line contains the pair of vertices' indexes, joined by an edge, a_i, b_i ($1 \le a_i, b_i \le n$). It's guaranteed that the graph described in the input is a tree.

Output

Output the only number — the maximum product of sizes of the connected components, that Bob can get after deleting some of the tree's edges.

Examples

input
5 1 2 2 3 3 4 4 5
output
6

input
8 1 2 1 3 2 4 2 5 3 6 3 7 6 8
output
18

input
3 1 2 1 3
output
3