








































# Practice Arena

Practice problems aimed to improve your coding skills.

-  PRACTICE-02\_SCAN-PRINT
-  PRACTICE-03\_TYPES
-  LAB-PRAC-02\_SCAN-PRINT
-  LAB-PRAC-01
-  PRACTICE-04\_COND
-  BONUS-PRAC-02
-  LAB-PRAC-03\_TYPES
-  PRACTICE-05\_COND-LOOPS
-  LAB-PRAC-04\_COND
-  LAB-PRAC-05\_CONDLLOOPS
-  PRACTICE-07\_LOOPS-ARR
-  LAB-PRAC-06\_LOOPS
-  LAB-PRAC-07\_LOOPS-ARR
-  LABEXAM-PRAC-01\_MIDSEM
-  PRACTICE-09\_PTR-MAT
-  LAB-PRAC-08\_ARR-STR
-  PRACTICE-10\_MAT-FUN
-  LAB-PRAC-09\_PTR-MAT
-  LAB-PRAC-10\_MAT-FUN
-  PRACTICE-11\_FUN-PTR
-  LAB-PRAC-11\_FUN-PTR
-  LAB-PRAC-12\_FUN-STRUC
  -  Point Pairing Party
  -  Verify the family tree of Mr C
  -  Simple Sudoku
  -  The Family Tree of Mr C Part Three
  -  The Post offices of KRville
  -  Matrix Mandala
  -  Mango Mania
  -  Recover the Rectangle
  -  Crazy for Candy
  -  A Brutal Cipher Called Brutus
  -  Triangle Tangle
  -  Basic Balanced Bracketing
-  LABEXAM-PRAC-02\_ENDSEM
-  LAB-PRAC-13\_STRUC-NUM
-  LAB-PRAC-14\_SORT-MISC

# Triangle Tangle

## LAB-PRAC-12\_FUN-STRUC

**Triangle Tangle [20 marks]**

---

**Problem Statement**

In the first line of the input we will give you  $n$ , a strictly positive integer. In the next  $n$  lines, we will give you the three corners of  $n$  triangles on the 2D plane. Each corner will be described using its  $x$  and  $y$  coordinates. All coordinates will be integers. To specify a triangle, we will give you 6 integer coordinates, separated by a space as described below

$x_1 y_1 x_2 y_2 x_3 y_3$

This will describe a triangle. In the last line of the input, we will give you two points on the 2D plane which will uniquely define a line (as well as a line segment). Both points will be similarly give to you by specifying their  $x$  and  $y$  coordinates, separated by a space. All coordinates will be integers.

$x_1 y_1 x_2 y_2$

In your output you have to print  $n$  lines, telling us which all of the triangles intersect the line. If a certain triangle does intersect with the line, print "YES" (without quotes) else print "NO" (without quotes). An intersection is counted whenever a triangle and a line share more than one point in common. Thus, if only one vertex of the triangle lies on the line, it is not counted as an intersection. However, if an entire edge of the triangle is present on the line, it is counted as an intersection.

**Caution**

1. The question may require you to compare floating point numbers like slopes etc. Since comparing floating point numbers for equality is dangerous, we will consider two floating point numbers to be the same if their difference in absolute terms is less than 0.0001. Use the `fabs()` function from `math.h` to get the absolute value of floating point numbers.
2. Be careful that coordinates can be negative too. Also be careful that the line we give you may be a vertical or a horizontal line. Also be careful that the line we give you may actually be along one of the edges of one of the triangles.
3. We will never give you line segments with zero length i.e. the two points in the last line will never be the same.
4. Be careful about extra/missing lines and extra/missing spaces in your output.

**HINTS:** You may want to use structures (although it is not compulsory to use structures) to make your code neater and easier to debug. As we saw in class, create a structure

```
struct Point{
    int x,y;
};
using which you can create line structures
struct Line{
    struct Point p1, p2;
};
and triangle structures
struct Triangle{
    struct Point p1, p2, p3;
};
```

---

**EXAMPLE:**

INPUT

2

0 0 2 0 0 2

5 6 7 8 9 11

0 0 1 1

OUTPUT:

YES

NO

---

**Grading Scheme:**Total marks: **[20 Points]**

There will be partial grading in this question. There are several lines in your output. Printing each line correctly, in the correct order, carries equal weightage. Each visible test case is worth 2 points and each hidden test case is worth 4 points. There are 2 visible and 4 hidden test cases.

Please remember, however, that when you press Submit/Evaluate, you will get a green bar only if all parts of your answer are correct. Thus, if your answer is only partly correct, Prutor will say that you have not passed that test case completely, but when we do autograding afterwards, you will get partial marks.

 **Start Solving! (/editor/practice/6240)**