
Point in interval**X30452_en**

Write a function that, given a vector of n closed intervals in the real line, say $I = [[a_0, b_0], [a_1, b_1], \dots, [a_{n-1}, b_{n-1}]]$ with $a_i \leq b_i$, and a real value x , returns true if, and only if, x is contained in some interval. The function must not examine all intervals if there's no need to do so. The input of the program that uses this function will have several cases for x . You must use the following C++ code, completing the missing parts, and not changing anything else:

```
#include <iostream>
#include <vector>
using namespace std;

struct interval {
    double a, b;
};
typedef vector<interval> set_of_intervals;

// reads n intervals from cin, given as a0 b0 a1 b1 ...
// see the sample inputs
set_of_intervals read_intervals(int n) {
    set_of_intervals I(n);
    ...
    return I;
}

bool inside_some_interval(double x, ...) {
    ...
}

int main() {
    int n;
    cin >> n;
    set_of_intervals I = read_intervals(n);
    int m;
    cin >> m;
    for (int i = 0 ; i < m ; ++i) {
        double x; cin >> x;
        if (inside_some_interval(x, I))
            cout << "yes" << endl;
        else
            cout << "no" << endl;
    }
}
```

Exam score: 2.5 Automatic part: 100%

Input

The input has the number n of intervals in the first line. The second line has the description of the intervals: $2n$ reals $a_0 b_0 a_1 b_1 \dots a_{n-1} b_{n-1}$, such that $a_i \leq b_i$ for all i . The third line has the number m of cases for x . The fourth line has m reals x for which we want to know if they are in some of the given intervals.

Output

For each case the program outputs to cout the word yes if x is inside at least one of the given intervals, and the word no otherwise; the output for each case is always followed by a line break.

Sample input

```
3
1 2 1.5 3 -1 0
9
0.5 1.5 3.14159 -0.5 1 0 -8 15 0.9
```

Sample output

```
no
yes
no
yes
yes
yes
no
no
no
```

Problem information

Author : PRO1

Generation : 2015-11-21 13:55:42

© Jutge.org, 2006–2015.

<http://www.jutge.org>