**BDAT 1004 – Data Programming**

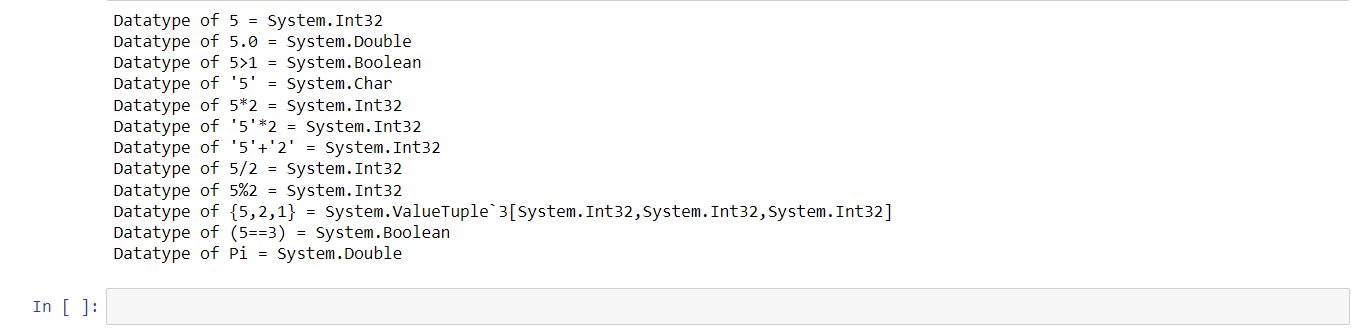
**C# - Visual Studio Outputs:**

**Question: 1**

|  |
| --- |
| What data type is each of the following?  5 |
| 5.0 |
| 5 > 1 |
| '5' |
| 5 \* 2 |
| '5' \* 2 |
| '5' + '2' |
| 5 / 2 |
| 5 %2 |
| {5, 2, 1} |
| 5 == 3 |
| Pi (the number) |

**Answer:**



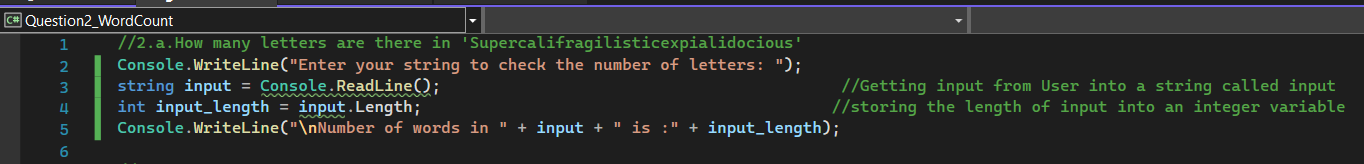


**Question :2**

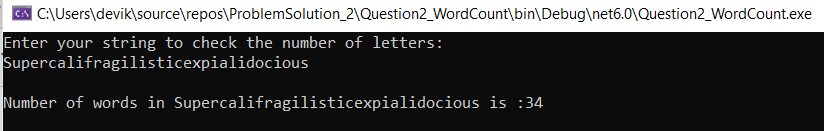
Write (and evaluate) C# expressions that answer these questions:

1. a. How many letters are there in 'Supercalifragilisticexpialidocious'?

**Code:**

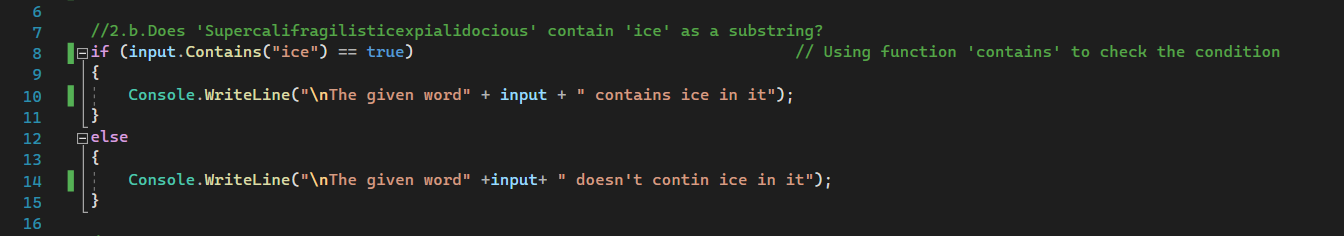


**Output:**

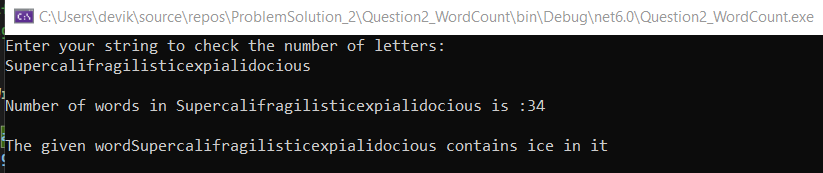


1. b. Does 'Supercalifragilisticexpialidocious' contain 'ice' as a substring?

**Code:**

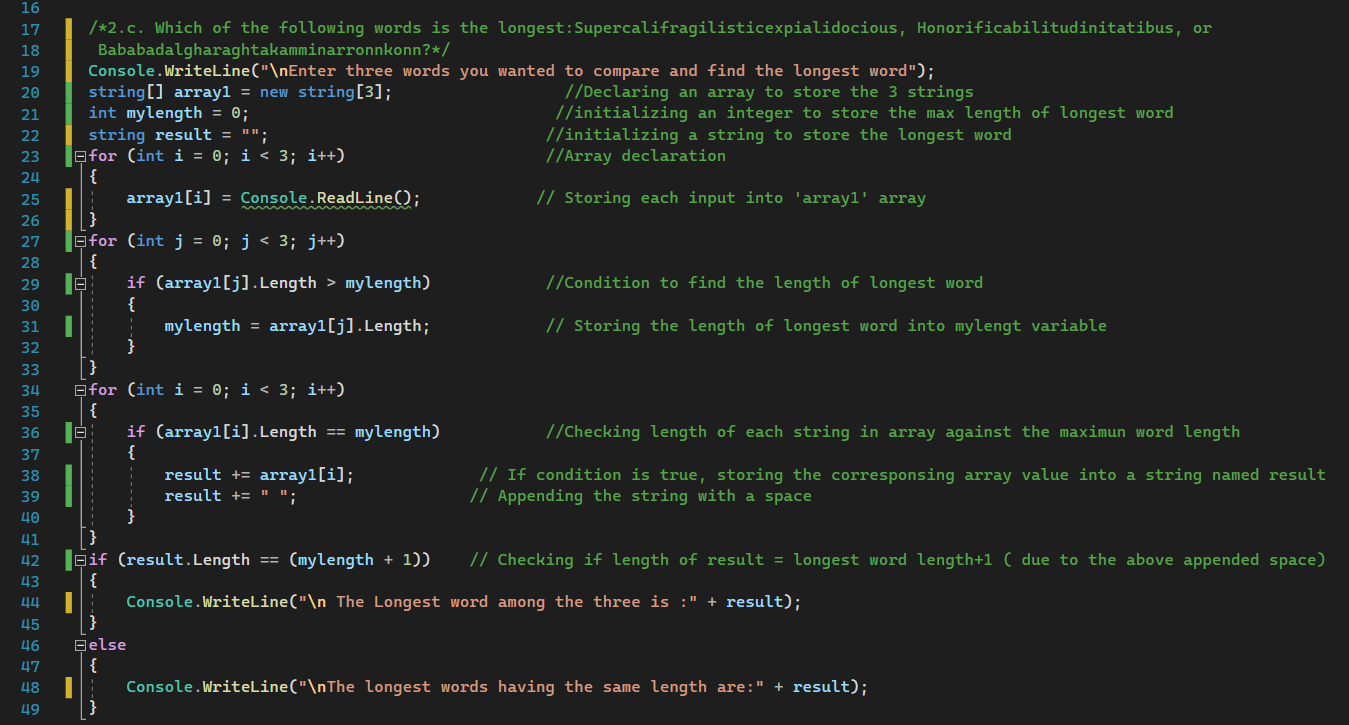


**Output:**

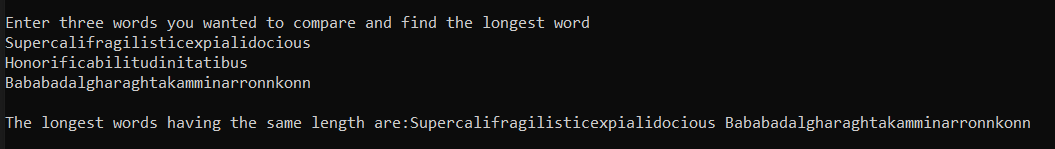


1. c. Which of the following words is the longest: Supercalifragilisticexpialidocious, Honorificabilitudinitatibus, or Bababadalgharaghtakamminarronnkonn?

**Code:**

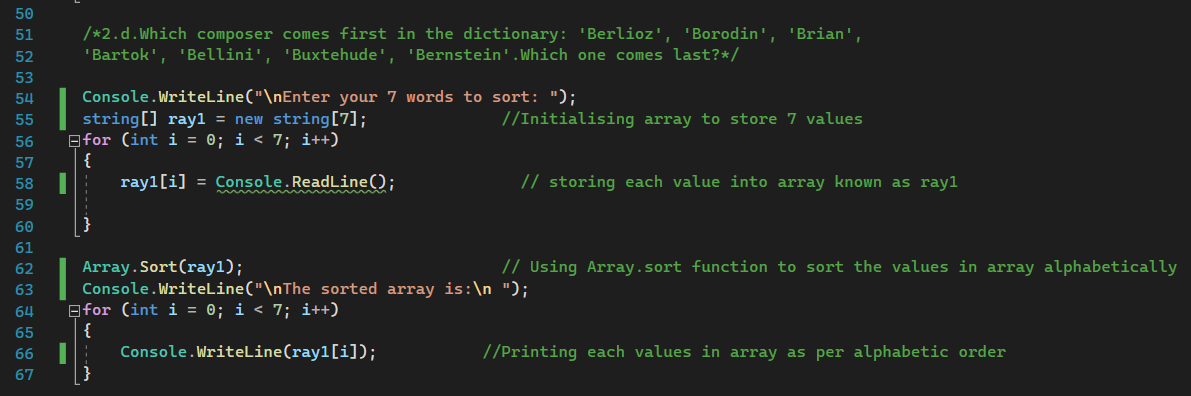


**Output:**



1. d. Which composer comes first in the dictionary: 'Berlioz', 'Borodin', 'Brian', 'Bartok', 'Bellini', 'Buxtehude', 'Bernstein'. Which one comes last?

**Code:**



**Output:**



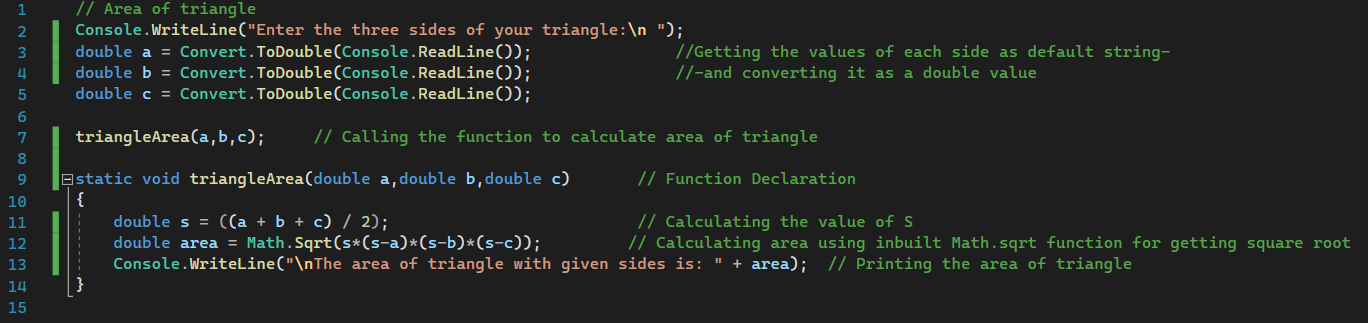
**Question :3**

Implement function triangleArea(a,b,c) that takes as input the lengths of the 3 sides of a triangle and returns the area of the triangle. By Heron's formula, the area of a triangle with side lengths a, b, and c is , where . s(s - a)(s -b)(s -c)s = (a +b + c) /2

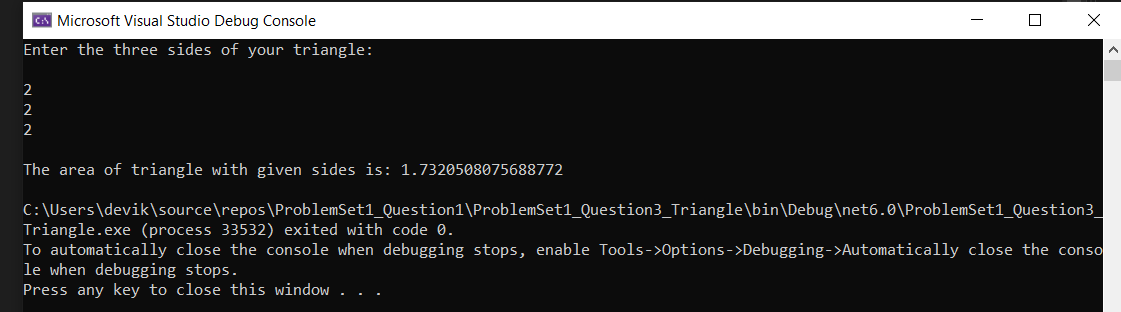
>>> triangleArea(2,2,2)

1.7320508075688772

**Code:**



**Output:**



**Question :4**

Write a program in C# Sharp to separate odd and even integers in separate arrays. Go to the editor

Test Data :

Input the number of elements to be stored in the array :5

Input 5 elements in the array :

element - 0 : 25

element - 1 : 47

element - 2 : 42

element - 3 : 56

element - 4 : 32

Expected Output:

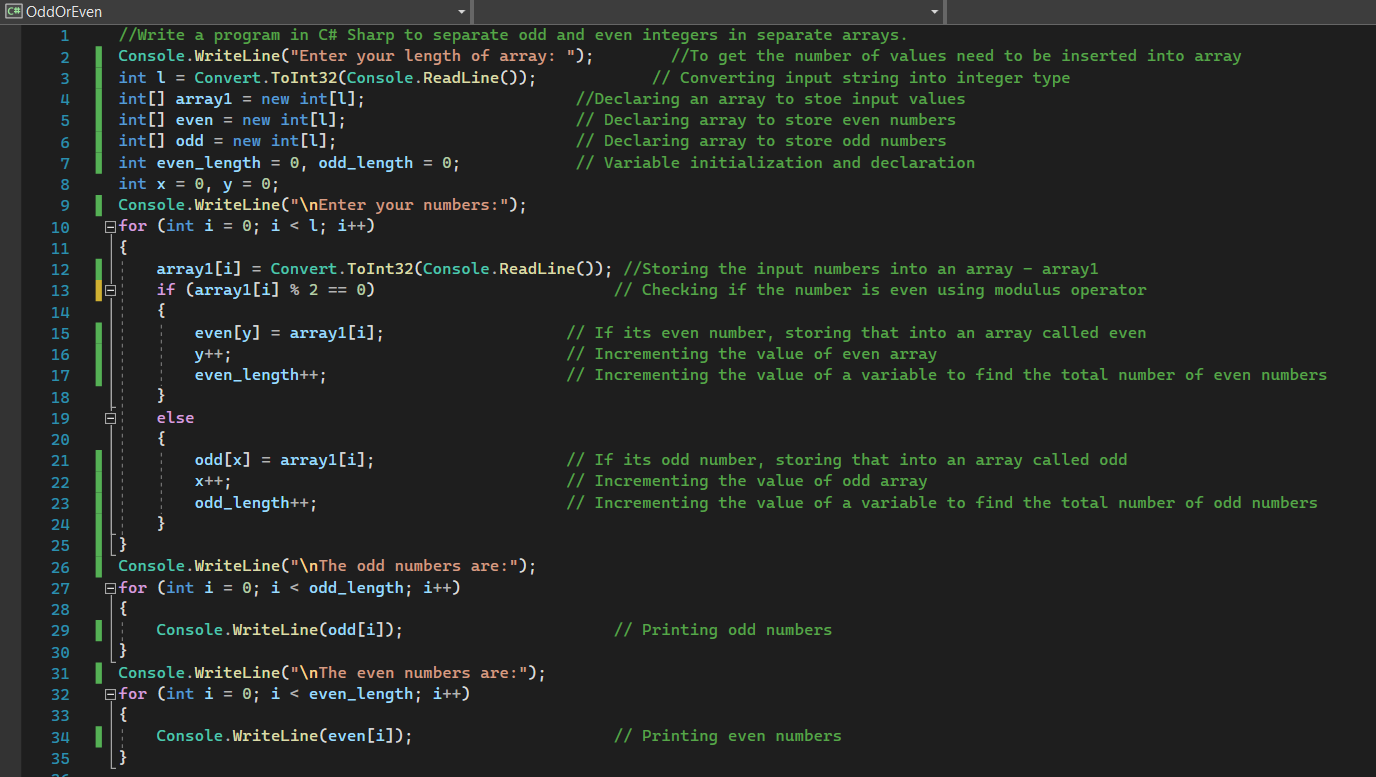
The Even elements are:

42 56 32

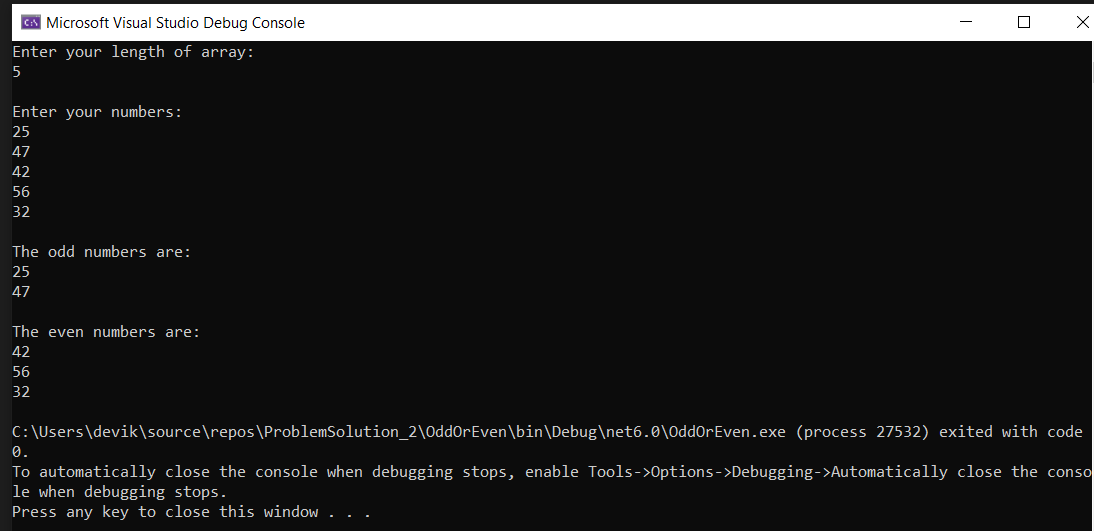
The Odd elements are :

25 47

**Code:**



**Output:**



**Question :5**

1. a. Write a function inside(x,y,x1,y1,x2,y2) that **returns** True or False depending on whether the point (x,y) lies in the rectangle with lower left corner (x1,y1) and upper right corner (x2,y2).

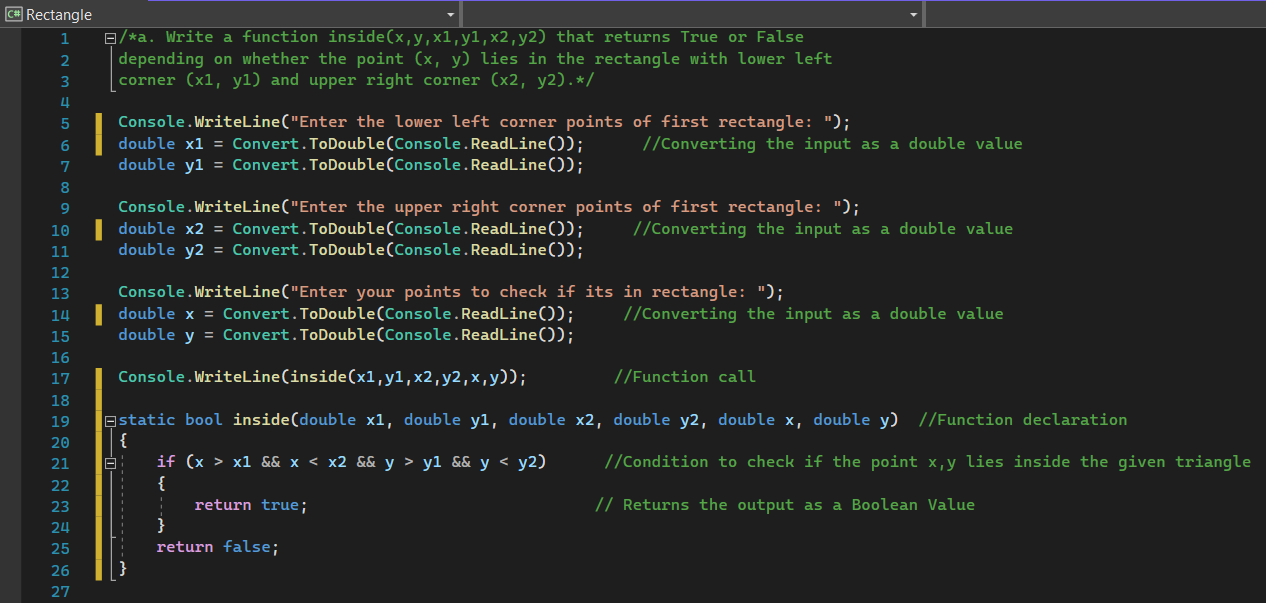
>>> inside(1,1,0,0,2,3)

True

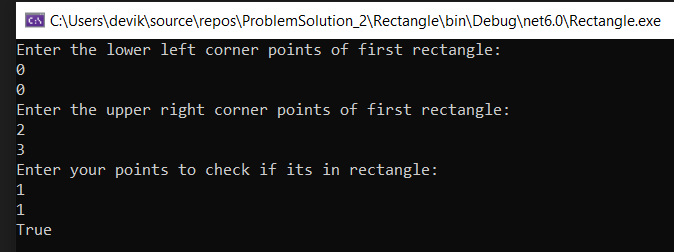
>>> inside(-1,-1,0,0,2,3)

False

**Code:**

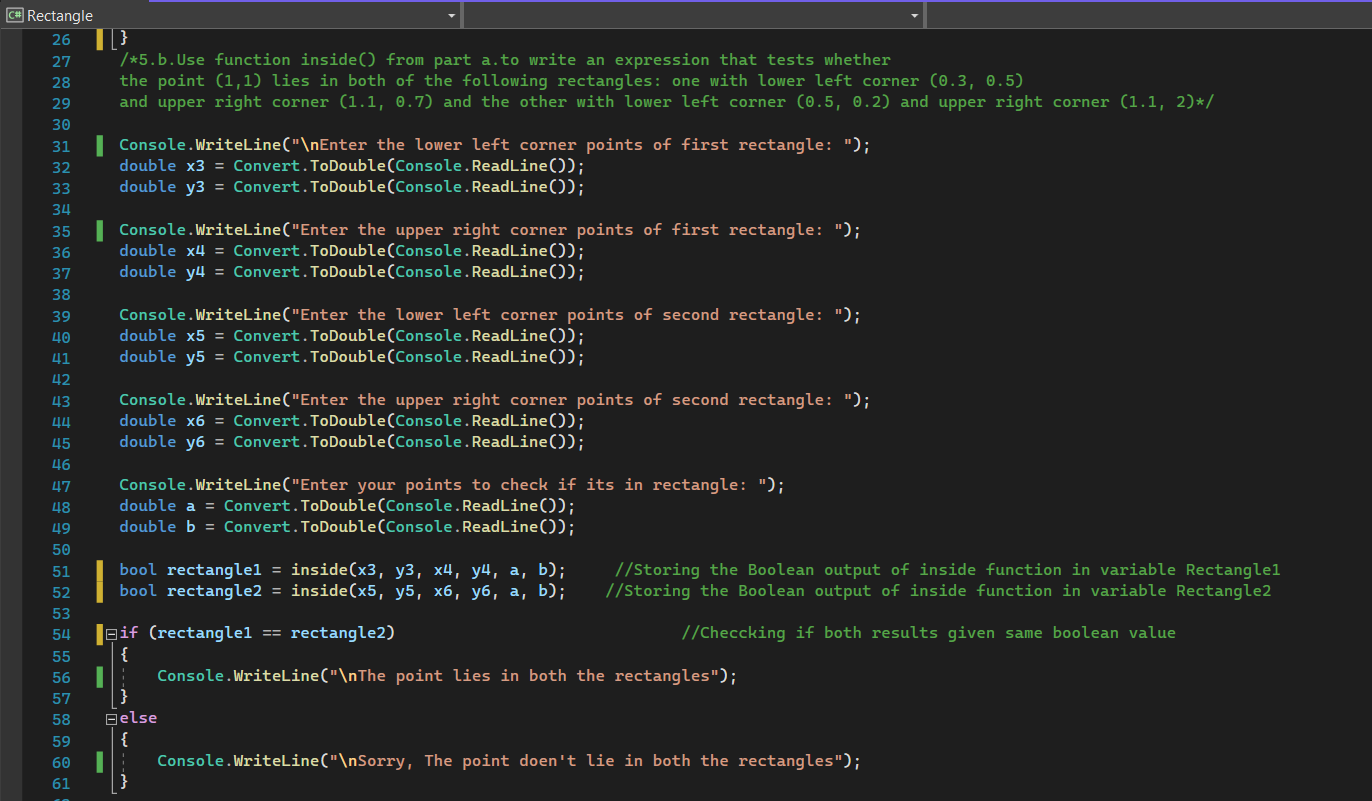


**Output:**



1. b. Use function inside() from part a. to write an expression that tests whether the point (1,1) lies in both of the following rectangles: one with lower left corner (0.3, 0.5) and upper right corner (1.1, 0.7) and the other with lower left corner (0.5, 0.2) and upper right corner (1.1, 2).

**Code:**



**Output:**

