**Q1**

5.0 is a float, meaning it is a number with a decimal point.

5 > 1 is a boolean expression, meaning it is either true or false.

'5' is a string, meaning it is a sequence of characters.

5 \* 2 is an integer, meaning it is a whole number.

'5' \* 2 is a string, meaning it is a sequence of characters.

'5' + '2' is a string, meaning it is a sequence of characters.

5 / 2 is a float, meaning it is a number with a decimal point.

5 % 2 is an integer, meaning it is a whole number.

{5, 2, 1} is a set, meaning it is a collection of unique items.

5 == 3 is a boolean expression, meaning it is either true or false.

Pi (the number) is a float, meaning it is a number with a decimal point.

**Q2 a**

using System;

namespace ProblemSet

{

class HelloWorld

{

static void Main()

{

string str = "Supercalifragilisticexpialidocious";

Console.WriteLine("Number of Letters: " + str.Length);

}

}

}

}

**Q2 b**

using System;

namespace ProblemSet

{

class HelloWorld

{

static void Main()

{

string str = "Supercalifragilisticexpialidocious";

//index of return starting index of given substring if string does not contain it then it return -1

int charPos = str.IndexOf("ice");

if (charPos == -1)

{

Console.WriteLine("String does not contain 'ice' as a substring.");

}

else

{

Console.WriteLine("String contain 'ice' as a substring.");

}

}

}

}

**Q3**

using System;

namespace ProblemSet

{

class HelloWorld

{

static double triangleArea(int a, int b, int c) // No Parameter

{

// calculate the value of s

int s = (a + b + c) / 2;

// calculate the area using Heron's formula

double area = Math.Sqrt(s \* (s - a) \* (s - b) \* (s - c));

// return the area calculated

return area;

}

}

}

**Q4**

using System;

namespace ProblemSet

{

class OddEven

{

static void Main()

{

// take the array size and declare array

Console.Write("Input the number of elements to be stored in the array: ");

int n = Convert.ToInt32(Console.ReadLine());

int[] arr = new int[n];

// input the array elements

Console.WriteLine("Input " + n + " elements in the array");

for (int i = 0; i < n; i++)

{

Console.Write("element - " + i + " : ");

arr[i] = Convert.ToInt32(Console.ReadLine());

}

// segregate the odd and even elements

int[] even = new int[n];

int[] odd = new int[n];

int evenIndex = 0, oddIndex = 0;

for (int i = 0; i < n; i++)

{

if (arr[i] % 2 == 0)

{

even[evenIndex] = arr[i];

evenIndex++;

}

else

{

odd[oddIndex] = arr[i];

oddIndex++;

}

}

// display the odd and even elements

Console.WriteLine("The even elements are:");

for (int i = 0; i < evenIndex; i++)

Console.Write(even[i] + " ");

Console.WriteLine("\nThe odd elements are:");

for (int i = 0; i < oddIndex; i++)

Console.Write(odd[i] + " ");

}

}

}

**Q5**

using System;

namespace ProblemSet

{

class GFG

{

// function to find if given point lies inside a given rectangle or not.

static bool inside(int x, int y, int x1, int y1, int x2, int y2)

{

if (x > x1 && x < x2 && y > y1 && y < y2)

return true;

return false;

}

public static void Main()

{

// bottom-left and top-right corners of rectangle

int x1 = 0, y1 = 0, x2 = 10, y2 = 8;

// given point

int x = 1, y = 5;

// function call

if (inside(x, y, x1, y1, x2, y2))

Console.Write("Yes");

else

Console.Write("No");

}

}

}