

namespace Assignment2

{

public delegate int MathOperation (int x, int y);

class MathOperations

{

public int Sum(int x, int y)

{

return x + y;

}

public int Product(int x, int y)

{

return x \* y;

}

static void Main(string[] args)

{

MathOperations ops = new MathOperations();

MathOperation sum = ops.Sum;

MathOperation prod = ops.Product;

MathOperation division = (x, y) => x / y;

int total = sum(10, 20);

int multi = prod(10, 20);

int div = division(40, 20);

Console.WriteLine("sum of 10,20 = " + total);

Console.WriteLine("multi of 10,20 = " + multi);

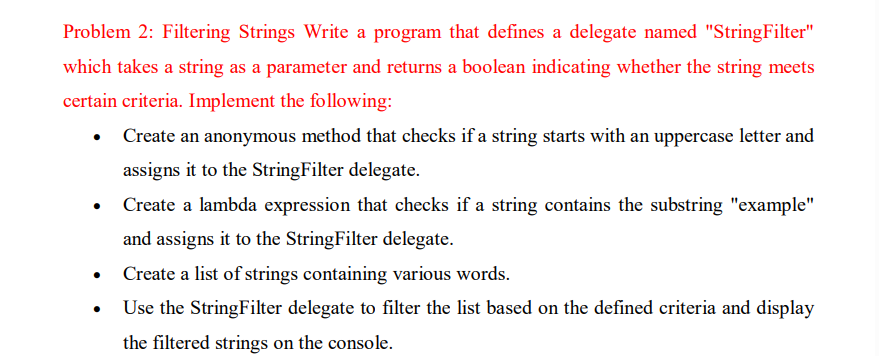
Console.WriteLine("div of 40,20 = " + div);

//Console.WriteLine("Hello, World!");

}

}

}



using System;

namespace Que2

{

//public delegate bool StringFilter(string str);

class Program

{

static void Main(string[] args)

{

Predicate<string> StringFilter = x =>

{

char s = x[0];

if(char.IsUpper(s) && x.Contains("example"))

return true;

return false;

};

List<string> StringList = new List<string>()

{

"Anand",

"Example",

"bestExample",

"niceExample"

};

foreach(string s in StringList)

{

if(StringFilter(s))

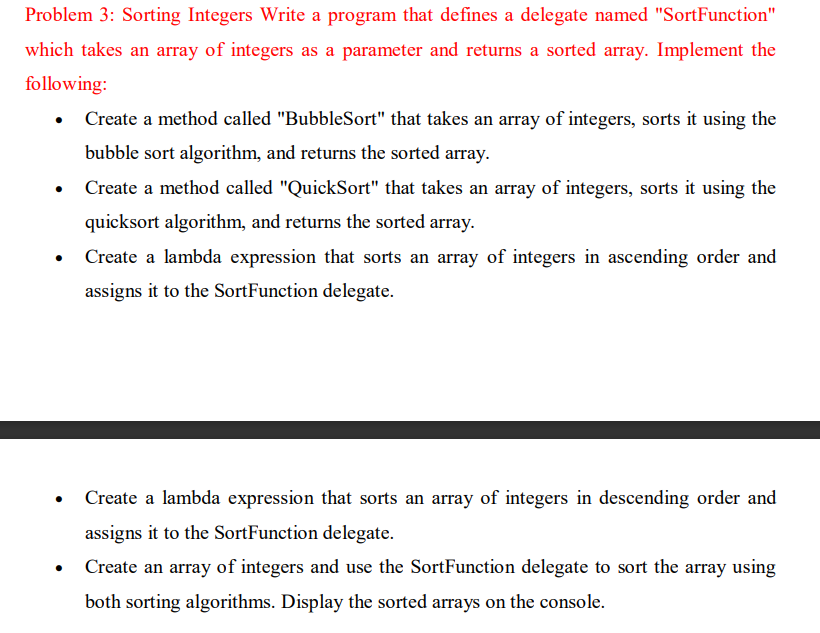
Console.WriteLine(s);

}

}

}

}



namespace Problem3Assignment2

{

internal class Program

{

//public delegate int[] SortFunction(int[] arr);

public static int[] BubbleSort(int[] arr)

{

for (int i = 0; i < arr.Length; i++)

{

for (int j = i + 1; j < arr.Length; j++)

{

if (arr[i] > arr[j])

{

int temp = arr[i];

arr[i] = arr[j];

arr[j] = temp;

}

}

}

return arr;

}

public static int[] BubbleSortDescending(int[] arr)

{

for (int i = 0; i < arr.Length; i++)

{

for (int j = i + 1; j < arr.Length; j++)

{

if (arr[i] < arr[j])

{

int temp = arr[i];

arr[i] = arr[j];

arr[j] = temp;

}

}

}

//Array.Reverse(arr);

return arr;

}

public static void Display(int[] arr)

{

foreach (int i in arr)

Console.Write(i + " ");

Console.WriteLine();

}

public static int[] SortArray(int[] array, int leftIndex, int rightIndex)

{

var i = leftIndex;

var j = rightIndex;

var pivot = array[leftIndex];

while (i <= j)

{

while (array[i] < pivot)

{

i++;

}

while (array[j] > pivot)

{

j--;

}

if (i <= j)

{

int temp = array[i];

array[i] = array[j];

array[j] = temp;

i++;

j--;

}

}

if (leftIndex < j)

SortArray(array, leftIndex, j);

if (i < rightIndex)

SortArray(array, i, rightIndex);

return array;

}

static void Main(string[] args)

{

int[] arr = { 10, 5, 20, 65, 30, 50, 60 };

Func<int[], int[]> SortFunction = BubbleSort;

Func<int[], int[]> SortDescFunction = BubbleSortDescending;

Func<int[], int, int, int[]> QuickSortFunction = SortArray;

int[] result = SortFunction(arr);

Display(result);

int[] result1 = SortDescFunction(arr);

Display(result1);

int[] result2 = QuickSortFunction(arr, 0, arr.Length - 1);

Display(result2);

}

}

}