

Федеральное агентство связи

**Государственное бюджетное образовательное учреждение высшего
образования**

Ордена Трудового Красного Знамени

«Московский технический университет связи и информатики»

Кафедра «МКиИТ»

дисциплина «СиАОД»

Отчет по Курсовой работе

Подготовил студент

группы БВТ1901: Нкурикийе Х.

Проверил: Мелехин А.

Москва 2021

Задания:

1. Массив A состоит из целых положительных чисел длин отрезков. Составьте из трех отрезков такой треугольник, чтобы его периметр был максимально возможным. Если невозможно составить треугольник с положительной площадью функция возвращает 0.

```
static int Task1(ref int[] arr)
{
    Array.Sort(arr);
    Array.Reverse(arr);
    for(int i = 0; i < arr.Length-2; i++)
    {
        if ((arr[i] < (arr[i+1] + arr[i+2])) && (arr[i+1] < (arr[i] + arr[i+2])) &&
            (arr[i+2] < (arr[i+1] + arr[i])))
        {
            return (arr[i] + arr[i+1] + arr[i+2]);
        }
    }
    return 0;
}
```

2. Дан массив неотрицательных целых чисел nums. Расположите их в таком порядке, чтобы вместе они образовали максимально возможное число.

```
static void Task2(ref int[] arr)
{
    int MatrixSize = arr.Length;
    int index = 0;

    for (int i = 0; i < MatrixSize - 1; i++)
    {
        index = i;

        for (int j = i + 1; j < MatrixSize; j++)
        {
            String str1 = Convert.ToString(arr[j]) + Convert.ToString(arr[index]);
            String str2 = Convert.ToString(arr[index]) + Convert.ToString(arr[j]);

            if (Convert.ToInt64(str1).CompareTo(Convert.ToInt64(str2)) == -1)
            {
                index = j;
            }
        }

        if (index != i)
        {
            int temp = arr[i];
            arr[i] = arr[index];
            arr[index] = temp;
        }
    }
}
```

3. Дана матрица `mat` размером $m * n$, значения целочисленные. Напишите функцию, сортирующую каждую диагональ матрицы по возрастанию и возвращающую получившуюся матрицу.

```
static void Task3(int[,] arr, int m,int n)
{
    for(int i = 0; i < n-1; i++)
    {
        FuncForTask3(arr, 0,i,m,n);
    }

    for (int i = 1; i < m - 1; i++)
    {
        FuncForTask3(arr, i, 0, m, n);
    }
}

static void FuncForTask3(int[,] arr, int m,int k,int lenX,int lenY)
{
    List<int> NewArr = new List<int>();

    int m1 = m;
    int k1 = k;
    while (ProvForTask3(m1,k1,lenX,lenY))
    {
        NewArr.Add(arr[m1, k1]);
        m1++;
        k1++;
    }

    NewArr.Sort();
    int g = 0;
    while (ProvForTask3(m, k, lenX, lenY))
    {
        arr[m,k]=NewArr[g];
        m++;
        k++;
        g++;
    }
}

static bool ProvForTask3(int indexX,int indexY, int i,int j)
{
    if (indexX < i && indexY < j) return true;
    else return false;
}
```

4. Дан массив отрезков `intervals`, в котором `intervals[i] = [starti , endi]`, некоторые отрезки могут пересекаться. Напишите функцию, которая объединяет все пересекающиеся отрезки в один и возвращает новый массив непересекающихся отрезков.

```

static void Task4(List<List<int>> MyList)
{
    for (int i = 0; i < MyList.Count - 1; i++)
    {
        for (int j = i + 1; j < MyList.Count; j++)
        {
            if ((MyList[i][0] >= MyList[j][0] && MyList[i][0] <= MyList[j][1]) ||
                (MyList[j][0] >= MyList[i][0] && MyList[j][0] <= MyList[i][1]))
            {
                List<int> arr = new List<int>();
                arr.Add(Math.Min(MyList[i][0], MyList[j][0]));
                arr.Add(Math.Max(MyList[i][1], MyList[j][1]));
                MyList[i] = arr;
                MyList.Remove(MyList[j]);
            }
        }
    }

    for(int i = 0; i < MyList.Count; i++)
    {
        Console.WriteLine(MyList[i][0] + "," + MyList[i][1] + " ");
    }
}

```

5. Дан массив points, где points[i] = [xstart, xend]. Напишите функцию, возвращающую минимальное количество стрел, которые нужно выпустить, чтобы уничтожить все шарики.

```

static void Task5(List<List<int>> MyList)
{
    for (int i = 0; i < MyList.Count-1; i++)
    {
        for (int j = i+1; j < MyList.Count; j++)
        {
            if((MyList[i][0]>= MyList[j][0] && MyList[i][0] <= MyList[j][1])||
                (MyList[j][0] >= MyList[i][0] && MyList[j][0] <= MyList[i][1]))
            {
                MyList.RemoveAt(j);
            }
        }
    }

    Console.WriteLine(MyList.Count);
}

```

6. На столе стоят $3n$ стопок монет. Вы и ваши друзья Алиса и Боб забираете стопки монет по следующему алгоритму:

1. Вы выбираете 3 стопки монет из оставшихся на столе.
2. Алиса забирает себе стопку с максимальным количеством монет.
3. Вы забираете одну из двух оставшихся стопок.
4. Боб забирает последнюю стопку.
5. Если еще остались стопки, то действия повторяются с первого шага. Дан

Напишите функцию, возвращающую максимальное число монет, которое вы можете получить

```
static void Task6(List<int> arr)
{
    arr.Sort();
    arr.Reverse();
    int sum = 0;
    int n = 0;
    int count = arr.Count;
    while (n!=count/3)
    {
        arr.RemoveAt(0);
        sum += arr[0];
        arr.RemoveAt(0);
        n++;
    }

    Console.WriteLine(sum);
}
```

7. Даны две строки: s1 и s2 с одинаковым размером, проверьте, может ли некоторая перестановка строки s1 “победить” некоторую перестановку строки s2 или наоборот. Строка x может “победить” строку y (обе имеют размер n), если $x[i] \geq y[i]$ (в алфавитном порядке) для всех i от 0 до n-1.

```
static void Task7(char[] ch1, char[] ch2)
{
    char[] ch3=ch1;
    char[] ch4=ch2;
    Array.Sort(ch1);
    Array.Sort(ch2);
    if(ch3==ch1)
    {
        char temp = ch1[ch1.Length-1];
        ch1[ch1.Length - 1] = ch1[ch1.Length - 2];
        ch1[ch1.Length - 2] = temp;
    }
    if (ch4 == ch2)
    {
        char temp = ch2[ch1.Length - 1];
        ch2[ch1.Length - 1] = ch2[ch1.Length - 2];
        ch2[ch1.Length - 2] = temp;
    }

    String st1 = "";
    String st2 = "";

    for(int i = 0; i < ch1.Length; i++)
    {
        st1 += ch1[i];
        st2 += ch2[i];
    }

    if (String.Compare(st1, st2) == 1)
    {
        Console.WriteLine("Ни одна перестановка второй строки не победит первую
строку");
    }
    else
    {

```

```

        Console.WriteLine("Вторая строка побеждает");
    }
}

```

8. Дана строка s, вернуть самую длинную полиндромную подстроку в s.

```

static void Task8(String str8)
{
    int n = 0;
    while (n != str8.Length)
    {
        String s = str8.Substring(0, str8.Length - n);
        if (IsPalindrom(s))
        {
            Console.WriteLine(s);
            return;
        }
        n++;
    }
}

static bool IsPalindrom(String str)
{
    bool result = true;
    int len = str.Length / 2;

    for(int i = 0; i < len; i++)
    {
        if (str[i] != str[str.Length - 1 - i]) return false;
    }

    return result;
}

```

9. Вернуть количество отдельных непустых подстрок текста, которые могут быть записаны как конкатенация некоторой строки с самой собой (т.е. она может быть записана, как $a + a$, где a - некоторая строка)

```

static void Task9(String str)
{
    int count = 0;
    List<String> strings = new List<string>();

    for (int i = 0; i < str.Length; i++)
    {
        int k = i;
        int z = 1;
        while (z < str.Length-i)
        {
            String Str = str.Substring(k, z);
            String Mystr = Str + Str;

            if (str.Contains(Mystr) && !strings.Contains(Mystr))
            {
                count++;
                strings.Add(Mystr);
            }

            z = z + 1;
        }
    }
}

```

```

    }

    Console.WriteLine(count);
}

```

Главная функция:

```

    Console.WriteLine("№1 Введите массив:");
    int[] MyArray1 = Array.ConvertAll(Console.ReadLine().Split(new char[] { ' ' }, StringSplitOptions.RemoveEmptyEntries), new Converter<String, int>(Convert.ToInt32));
    Console.WriteLine("Наибольший периметр треугольника: " + Task1(ref MyArray1));

    Console.WriteLine("№2 Введите массив:");
    int[] MyArray2 = Array.ConvertAll(Console.ReadLine().Split(new char[] { ' ' }, StringSplitOptions.RemoveEmptyEntries), new Converter<String, int>(Convert.ToInt32));

    Task2(ref MyArray2);
    Array.Reverse(MyArray2);
    for (int i = 0; i < MyArray2.Length; i++)
    {
        Console.Write(MyArray2[i]);
    }

    Console.ReadLine();

    int lines = 5;
    int columns = 6;
    int[,] array = CreateMatrix(lines, columns);
    Console.WriteLine("№3 Исходная матрица:");
    for (int i = 0; i < lines; i++)
    {
        for (int j = 0; j < columns; j++)
        {
            Console.Write(array[i, j] + " ");
        }
        Console.WriteLine("\r\n");
    }
    Task3(array, lines, columns);
    Console.WriteLine("Получившаяся матрица:");
    for (int i = 0; i < lines; i++)
    {
        for (int j = 0; j < columns; j++)
        {
            Console.Write(array[i, j] + " ");
        }
        Console.WriteLine("\r\n");
    }

    Console.WriteLine("№4 Введите массив отрезков:");

    int[] MyArray4 = Array.ConvertAll(Console.ReadLine().Split(new char[] { ' ' }, StringSplitOptions.RemoveEmptyEntries), new Converter<String, int>(Convert.ToInt32));
    List<List<int>> MyList4 = new List<List<int>>();
    for (int i = 0; i < MyArray4.Length; i = i + 2)
    {
        List<int> arr = new List<int>();
        arr.Add(MyArray4[i]);
        arr.Add(MyArray4[i + 1]);
        MyList4.Add(arr);
    }
    Task4(MyList4);
    Console.ReadLine();

    Console.WriteLine("№5 Введите массив шаров:");

```

```

        int[] MyArray5 = Array.ConvertAll(Console.ReadLine().Split(new char[] { ' ' }, StringSplitOptions.RemoveEmptyEntries), new Converter<String, int>(Convert.ToInt32));
        List<List<int>> MyList = new List<List<int>>>();
        for (int i = 0; i < MyArray5.Length; i = i + 2)
        {
            List<int> arr = new List<int>();
            arr.Add(MyArray5[i]);
            arr.Add(MyArray5[i + 1]);
            MyList.Add(arr);
        }
        Task5(MyList);

        Console.WriteLine("№6 Введите массив стопок:");
        int[] MyArray6 = Array.ConvertAll(Console.ReadLine().Split(new char[] { ' ', ',', ' ' }, StringSplitOptions.RemoveEmptyEntries), new Converter<String, int>(Convert.ToInt32));

        Task6(MyArray6.ToList<int>());

        Console.WriteLine("№7 Введите строку:");
        String str71 = Console.ReadLine();
        Console.WriteLine("Введите вторую строку:");
        String str72 = Console.ReadLine();

        char[] ch1 = new char[str71.Length];
        char[] ch2 = new char[str72.Length];

        for (int i = 0; i < str71.Length; i++)
        {
            ch1[i] = str71[i];
            ch2[i] = str72[i];
        }
        Task7(ch1, ch2);

        Console.WriteLine("№8 Введите строку:");
        Task8(Console.ReadLine());

        Console.WriteLine("№9 Введите строку:");
        Task9(Console.ReadLine());
        Console.ReadLine();

```


Результаты:

```
№1 Введите массив:
3 6 2 3
Наибольший периметр треугольника: 8
№2 Введите массив:
10 2
210
№3 Исходная матрица:
5341 5134 5793 5019 5257 5755

5590 5959 5265 5349 5050 5441

5338 5680 5725 5561 5307 5813

5745 5102 5832 5206 5048 5318

5770 5578 5874 5574 5114 5546

Получившаяся матрица:
5114 5048 5307 5019 5257 5755

5574 5206 5134 5318 5050 5441

5102 5590 5341 5265 5349 5813

5578 5338 5680 5725 5546 5793

5770 5745 5874 5832 5959 5561

№4 Введите массив отрезков:
1 4 4 5
1,5
№5 Введите массив шаров:
2 3 2 3
1
№6 Введите массив стопок:
2 4 5
4
```

```
№7 Введите строку:
abc
Введите вторую строку:
хуа
Вторая строка побеждает
№8 Введите строку:
babad
bab
№9 Введите строку:
abcsabcsabc
3
```

Вывод:

В результате проделанной работы были написаны алгоритмы для всех необходимых заданий.