

**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 Лабораторной работе №4**

Подготовил студент

группы БВТ1901: Балдова Татьяна

Проверил: Мелехин А.

Москва 2020

## Задания:

Реализовать следующие структуры данных:

- Стек (stack): операции для стека: инициализация, проверка на пустоту, добавление нового элемента в начало, извлечение элемента из начала;

- Дек (двусторонняя очередь, deque): операции для дека: инициализация, проверка на пустоту, добавление нового элемента в начало, добавление нового элемента в конец, извлечение элемента из начала, извлечение элемента из конца.

1. Отсортировать строки файла, содержащие названия книг, в алфавитном порядке с использованием двух деков

2. Дек содержит последовательность символов для шифровки сообщений. Дан текстовый файл, содержащий зашифрованное сообщение. Пользуясь деком, расшифровать текст. Известно, что при шифровке каждый символ сообщения заменялся следующим за ним в деке по часовой стрелке через один.

3. Даны три стержня и  $n$  дисков различного размера. Диски можно надевать на стержни, образуя из них башни. Перенести  $n$  дисков со стержня А на стержень С, сохранив их первоначальный порядок. При переносе дисков необходимо соблюдать следующие правила: - на каждом шаге со стержня на стержень переносить только один диск; - диск нельзя помещать на диск меньшего размера; - для промежуточного хранения можно использовать стержень В. Реализовать алгоритм, используя три стека вместо стержней А, В, С. Информация о дисках хранится в исходном файле.

4. Дан текстовый файл с программой на алгоритмическом языке. За один просмотр файла проверить баланс круглых скобок в тексте, используя стек.

5. Дан текстовый файл с программой на алгоритмическом языке. За один просмотр файла проверить баланс квадратных скобок в тексте, используя дек.

6. Дан файл из символов. Используя стек, за один просмотр файла напечатать сначала все цифры, затем все буквы, и, наконец, все остальные символы, сохраняя исходный порядок в каждой группе символов.

7. Дан файл из целых чисел. Используя дек, за один просмотр файла напечатать сначала все отрицательные числа, затем все положительные числа, сохраняя исходный порядок в каждой группе.

8. Дан текстовый файл. Используя стек, сформировать новый текстовый файл, содержащий строки исходного файла, записанные в обратном порядке: первая строка становится последней, вторая – предпоследней и т.д.

9. Дан текстовый файл. Используя стек, вычислить значение логического выражения, записанного в текстовом файле в следующей форме:  $\langle \text{ЛВ} \rangle ::= T \mid F \mid (N) \mid (A) \mid (X) \mid (O)$ , где буквами обозначены логические константы и операции: T – True, F – False, N – Not, A – And, X – Xor, O – Or

10. Дан текстовый файл. В текстовом файле записана формула следующего вида:  $::= M(,) \mid N(\text{Формула},) \mid \langle \text{Цифра} \rangle ::= 0 \mid 1 \mid 2 \mid 3 \mid 4 \mid 5 \mid 6 \mid 7 \mid 8 \mid 9$  где буквами обозначены функции: M – определение максимума, N – определение минимума. Используя стек, вычислить значение заданного выражения.

11. Дан текстовый файл. Используя стек, проверить, является ли содержимое текстового файла правильной записью формулы вида:  $\langle \text{Формула} \rangle ::= \langle \text{Терм} \rangle \mid \langle \text{Терм} \rangle + \langle \text{Формула} \rangle \mid \langle \text{Терм} \rangle - \langle \text{Формула} \rangle \mid \langle \text{Терм} \rangle ::= \langle \text{Имя} \rangle \mid (\langle \text{Формула} \rangle) \mid \langle \text{Имя} \rangle ::= x \mid y \mid z$

### Задание 1

```
static void Task1(List<String> books)
{
    MyDeque<String> q1 = new MyDeque<string>(books.Count + 1);
    MyDeque<String> q2 = new MyDeque<string>(books.Count + 1);

    for(int i = 0; i < books.Count; i++)
    {
        if (q1.Count == 0) { q1.PushBack(books[i]); }
        else if (String.Compare(books[i], q1.PeekBack()) == 1)
        {
            q1.PushBack(books[i]);
        }
        else if (String.Compare(books[i], q1.PeekFront()) == -1)
        {
            q1.PushFront(books[i]);
        }
        else
        {
            while(String.Compare(books[i], q1.PeekFront()) == 1)
            {
                q2.PushBack(q1.PopFront());
            }

            q1.PushFront(books[i]);

            while (q2.Count != 0)
            {
                q1.PushFront(q2.PopBack());
            }
        }
    }
}
```

```

        q1.PrintDec();
    }

```

## Задание 2

```

static void CreateDec2( )
{
    String Str = "абвгдежзиклмнопртсуфхцщщъьэюя";
    for (int i = 0; i < Str.Length; i++)
    {
        if(Char.IsLetter(Str[i]))
            MyDec2.PushBack(Str[i]);
    }
}

static Char Decode(char ch)
{
    for (int i = 0; i < MyDec2.Count; i++)
    {
        Char x = MyDec2.PopBack();
        if (ch == x)
        {
            MyDec2.PushFront(x);
            Char Value = MyDec2.PopBack();
            MyDec2.PushFront(Value);
            return Value;
        }

        MyDec2.PushFront(x);
    }

    return ch;
}

```

## Задание 3

```

static void move(MyStack<int> a, MyStack<int> b)
{
    if (a.Count > 0 && b.Count == 0) b.Push(a.Pop());
    else if (a.Count == 0 && b.Count > 0) a.Push(b.Pop());
    else if (a.Count > 0 && b.Count > 0 )
    {
        if (a.Peek() < b.Peek()) b.Push(a.Pop());
        else a.Push(b.Pop());
    }
}

```

## Задание 4

```

static void Task4(String str)
{
    MyStack<char> Stack3 = new MyStack<char>(str.Length);

    for(int i = 0; i < str.Length; i++)
    {
        if (str[i] == '(')
        {
            Stack3.Push('(');
        }
        else if(str[i] == ')')
        {
            if (Stack3.Count == 0)
            {

```

```

        Console.WriteLine("Количество '(' не равно количеству скобок
        ')');
        return;
    }
    Stack3.Pop();
}

if (Stack3.Count == 0) Console.WriteLine("Количество '(' равно количеству
скобок ')");
else Console.WriteLine("Количество '(' не равно количеству скобок ')");
}

```

## Задание 5

```

static void Task5(String str)
{
    MyDeque<char> Deque3 = new MyDeque<char>(str.Length+1);

    for (int i = 0; i < str.Length; i++)
    {
        if (str[i] == '[')
        {
            Deque3.PushBack('[');
        }
        else if (str[i] == ']')
        {
            if (Deque3.Count == 0)
            {
                Console.WriteLine("Количество '[' не равно количеству скобок
                ']');
                return;
            }
            Deque3.PopBack();
        }
    }

    if (Deque3.Count == 0) Console.WriteLine("Количество '[' равно количеству
скобок ']");
    else Console.WriteLine("Количество '[' не равно количеству скобок ']");
}

```

## Задание 6

```

static void Task6(String str)
{
    MyStack<char> Digits = new MyStack<char>(str.Length);
    MyStack<char> Letters = new MyStack<char>(str.Length);
    MyStack<char> Others = new MyStack<char>(str.Length);

    for (int i = str.Length-1; i >= 0; i--)
    {
        if(str[i]>='0' && str[i] <= '9')
        {
            Digits.Push(str[i]);
        }
        else if ((str[i] >= 'a' && str[i] <= 'z') || (str[i] >= 'A' && str[i] <=
'Z') || (str[i] >= 'а' && str[i] <= 'я') || (str[i] >= 'А' && str[i] <= 'Я'))
        {
            Letters.Push(str[i]);
        }
        else
        {
            Others.Push(str[i]);
        }
    }

    while (Digits.Count > 0)
    {
        Console.Write(Digits.Pop());
    }

    while (Letters.Count > 0)
    {
        Console.Write(Letters.Pop());
    }

    while (Others.Count > 0)
    {
        Console.Write(Others.Pop());
    }
}

```

## Задание 7

```

static void Task7(int[] arr)
{
    MyDeque<int> Dec4 = new MyDeque<int>(arr.Length+1);

    for(int i =0; i < arr.Length; i++)
    {
        if (arr[i] < 0)
        {
            Dec4.PushFront(arr[i]);
        }
        else Dec4.PushBack(arr[i]);
    }

    for (int i=1;i<arr.Length;i++)
    {
        int x = Dec4.PopFront();
        if (x < 0)
        {
            Dec4.PushBack(x);
        }
        else
        {
            Dec4.PushFront(x);
        }
    }
}

```

```

        break;
    }
}

while (true)
{
    int x = Dec4.PopBack();
    if (x < 0) Console.Write(x + " ");
    else
    {
        Dec4.PushBack(x);
        break;
    }
}

while (Dec4.Count > 0)
{
    Console.Write(Dec4.PopFront()+" ");
}

}

```

## Задание 8

```

static void Task8(String[] arr)
{
    MyStack<String> Stack8 = new MyStack<String>(arr.Length+1);

    for(int i = 0; i < arr.Length; i++)
    {
        Stack8.Push(arr[i]);
    }

    while (Stack8.Count != 0)
    {
        Console.WriteLine(Stack8.Pop());
    }

}

```

## Задание 9

```

static bool Task9(MyStack<bool> Zn, MyStack<char> Ops)
{
    if (Zn.Count == 1)
    {
        return Zn.Pop();
    }
    else
    {
        String op = "XOAN";

        while (Ops.Count != 0)
        {
            char CurOp = Ops.Pop();

            if (CurOp == 'N')
            {
                bool res = !Zn.Pop();
                Zn.Push(res);
            }
            else if (CurOp == 'n')
            {
                char NextOp = Ops.Pop();

```

```

        bool one = Zn.Pop();
        bool two = Zn.Pop();
        bool result=true;

        if (NextOp == 'A') result = !(one & two);
        else if(NextOp == 'O') result = !(one | two);
        else if(NextOp == 'X') result = !(one ^ two);

        Zn.Push(result);
    }
    else if(op.Contains(Convert.ToString(CurOp)))
    {
        bool one = Zn.Pop();
        bool two = Zn.Pop();
        bool result = true;

        if (CurOp == 'A') result = one & two;
        else if (CurOp == 'O') result = one | two;
        else if (CurOp == 'X') result = one ^ two;

        Zn.Push(result);
    }
    else
    {
        bool one= Zn.Pop();
        bool two = Task9(Zn, Ops);
        bool result = true;

        if (CurOp == 'a') result = one & two;
        else if (CurOp == 'o') result = one | two;
        else if (CurOp == 'x') result = one ^ two;
        Zn.Push(result);
    }
}

return Zn.Pop();
}

}

```

## Задание 10

```

static int Task10(MyStack<char> St)
{
    if (St.Count == 1 && Char.IsDigit(St.Peek())) return St.Pop();
    else
    {
        int num1, num2, result;
        char op = St.Pop();
        St.Pop();// убираем скобку перед операцией

        if (Char.IsDigit(St.Peek()))
        {
            int c = (int)St.Pop() - (int)'0';
            num1 = c;
            St.Pop();
        }
        else
        {
            num1 = Task10(St);
        }
    }
}

```



```

        if (Char.IsDigit(St.Peek()))
        {
            int c = (int)St.Pop() - (int)'0';
            num2 = c;
            St.Pop();
        }
        else num2 = Task10(St);

        int min, max;

        if (num1 > num2)
        {
            max = num1;
            min = num2;
        }
        else
        {
            max = num2;
            min = num1;
        }

        if (op == 'M')
        {
            return max;
        }
        else
        {
            return min;
        }
    }
}

```

## Задание 11

```

static void Task11(MyStack<char> St)
{
    String op = "+-";
    String chars = "xyz";

    while (St.Count != 0)
    {
        char ch = St.Pop();

        if (ch == '(' && St.Count >= 4){

            char ch1 = St.Pop();
            char ch2 = St.Pop();
            char ch3 = St.Pop();
            char ch4 = St.Pop();

            if(!(chars.Contains(Convert.ToString(ch1)) &&
op.Contains(Convert.ToString(ch2)) && chars.Contains(Convert.ToString(ch3)) && ch4 ==
'))))
            {
                Console.WriteLine("Является ли данное выражение формулой : " +
false);
            }
        }
    }
}

```

```

        return;
    }
}
else if(ch == '(' && St.Count < 4)
{
    Console.WriteLine("Является ли данное выражение формулой : " +
false);
    return;
}

else
{
    String prov = "";
    prov += ch;

    while(St.Count != 0 )
    {
        char ch1 = St.Pop();
        prov += ch1;

        if (ch1 == '(' || St.Count==0)
        {
            if (ch1 == '(')
            {
                prov = prov.Substring(0, prov.Length - 1);
                St.Push(ch1);
            }

            if (prov.Length != 0)
            {
                if (!(prov.Length == 2 &&
(chars.Contains(Convert.ToString(prov[1])) && op.Contains(Convert.ToString(prov[0])))) ||
chars.Contains(Convert.ToString(prov[0])) && op.Contains(Convert.ToString(prov[1])))) &&
!(prov.Length == 1 && op.Contains(Convert.ToString(prov[0]))))
                {
                    Console.WriteLine("Является ли данное выражение
формулой : " + false);
                    return;
                }
            }

            break;
        }
        else
        {
            if (!chars.Contains(Convert.ToString(ch1)) &&
!op.Contains(Convert.ToString(ch1)))
            {
                Console.WriteLine("Является ли данное выражение формулой
: " + false);
                return;
            }
        }
    }
}

Console.WriteLine("Является ли данное выражение формулой : " + true);
}

```

## Код вызова функций:

```
static void Main(string[] args)
{
    Console.WriteLine("Задача №1 ");
    List<String> Books = new List<string>();
    using (StreamReader sr = new StreamReader("task1.txt"))
    {
        for (int i = 0; i < size; i++)
        {
            Books.Add(sr.ReadLine());
        }
    }
    Task1(Books);

    Console.WriteLine("\r\n" + "Задача №2" );
    using (StreamReader sr = new StreamReader("task2.txt"))
    {
        String MyStr2 = sr.ReadToEnd();
        CreateDec2();
        for (int i = 0; i < MyStr2.Length; i++)
        {
            Console.Write(Decode(MyStr2[i]));
        }
    }

    Console.WriteLine("\r\n" + "Задача №3");
    int[] Disks;
    using (StreamReader sr = new StreamReader("task3.txt"))
    {
        Disks = Array.ConvertAll(sr.ReadToEnd().Split(new char[] { ' ' },
StringSplitOptions.RemoveEmptyEntries), new Converter<String, int>(Convert.ToInt32));
    }

    MyStack<int> Stack3A = new MyStack<int>(Disks.Length + 1);
    MyStack<int> Stack3B = new MyStack<int>(Disks.Length + 1);
    MyStack<int> Stack3C = new MyStack<int>(Disks.Length + 1);

    for (int i = Disks.Length-1; i >= 0; i--)
    {
        Stack3A.Push(Disks[i]);
    }

    while (Stack3C.Count != Disks.Length)
    {
        move(Stack3A, Stack3B);
        move(Stack3A, Stack3C);
        move(Stack3B, Stack3C);
    }

    while (Stack3C.Count != 0)
    {
        Console.WriteLine(Stack3C.Pop());
    }

    Console.WriteLine("\r\n" + "Задача №4");
    String Str4 = "";
    using (StreamReader sr = new StreamReader("task4.txt"))
    {
        Str4 = sr.ReadToEnd();
    }
    Task4(Str4);
}
```

```

Console.WriteLine("\r\n" + "Задача №5");
String Str5 = "";
using (StreamReader sr = new StreamReader("task5.txt"))
{
    Str5 = sr.ReadToEnd();
}
Task55(Str5);

Console.WriteLine("\r\n" + "Задача №6" );
String Str6 = "";
using (StreamReader sr = new StreamReader("task6.txt"))
{
    Str6 = sr.ReadToEnd();
}
Task6(Str6);

Console.WriteLine("\r\n" + "\r\n" + "Задача №7" );
int[] Numbers;
using (StreamReader sr = new StreamReader("task7.txt"))
{
    Numbers = Array.ConvertAll(sr.ReadToEnd().Split(new char[] { ' ' },
StringSplitOptions.RemoveEmptyEntries), new Converter<String, int>(Convert.ToInt32));
    Task7(Numbers);
}

Console.WriteLine("\r\n" + "\r\n" + "Задача №8" );
String[] Strings;
using (StreamReader sr = new StreamReader("task8.txt"))
{
    Strings = sr.ReadToEnd().Split(new char[] { ' ' },
StringSplitOptions.RemoveEmptyEntries);
}
Task8(Strings);

Console.WriteLine("\r\n" + "Задача №9");
String String9 = "";
using (StreamReader sr = new StreamReader("task9.txt"))
{
    String9= sr.ReadToEnd();
}

String op = "XOAN";
MyStack<bool> Zn = new MyStack<bool>(String9.Length);
MyStack<char> Ops = new MyStack<char>(String9.Length);

for (int i = String9.Length - 1; i >= 0; i--)
{
    if (op.Contains(Convert.ToString(String9[i])))
    {
        if (String9[i + 1] == '(') Ops.Push(Char.ToLower(String9[i]));
        else Ops.Push(String9[i]);
    }
    else if (String9[i] == 'T')
    {
        Zn.Push(true);
    }
    else if (String9[i] == 'F')
    {
        Zn.Push(false);
    }
}

Console.WriteLine(!(true & false));
Console.WriteLine(Task9(Zn, Ops));

```

```

Console.WriteLine("\r\n" + "Задача №10");
String String10 = "";
using (StreamReader sr = new StreamReader("task10.txt"))
{
    String10 = sr.ReadToEnd();
}
MyStack<char> St10 = new MyStack<char>(String10.Length+1);
if (String10[0] == '(' && String10[String10.Length - 1] == ')') String10 =
String10.Substring(1, String10.Length - 2);

for(int i = 0; i < String10.Length; i++)
{
    if(String10[i]==',' && (String10[i-1] == ')') || String10[i + 1] == '('))
    {
        String str1 = String10.Substring(0, i);
        String str2 = String10.Substring(i+1, String10.Length-i-1);
        String10 = str1 + str2;
    }
}

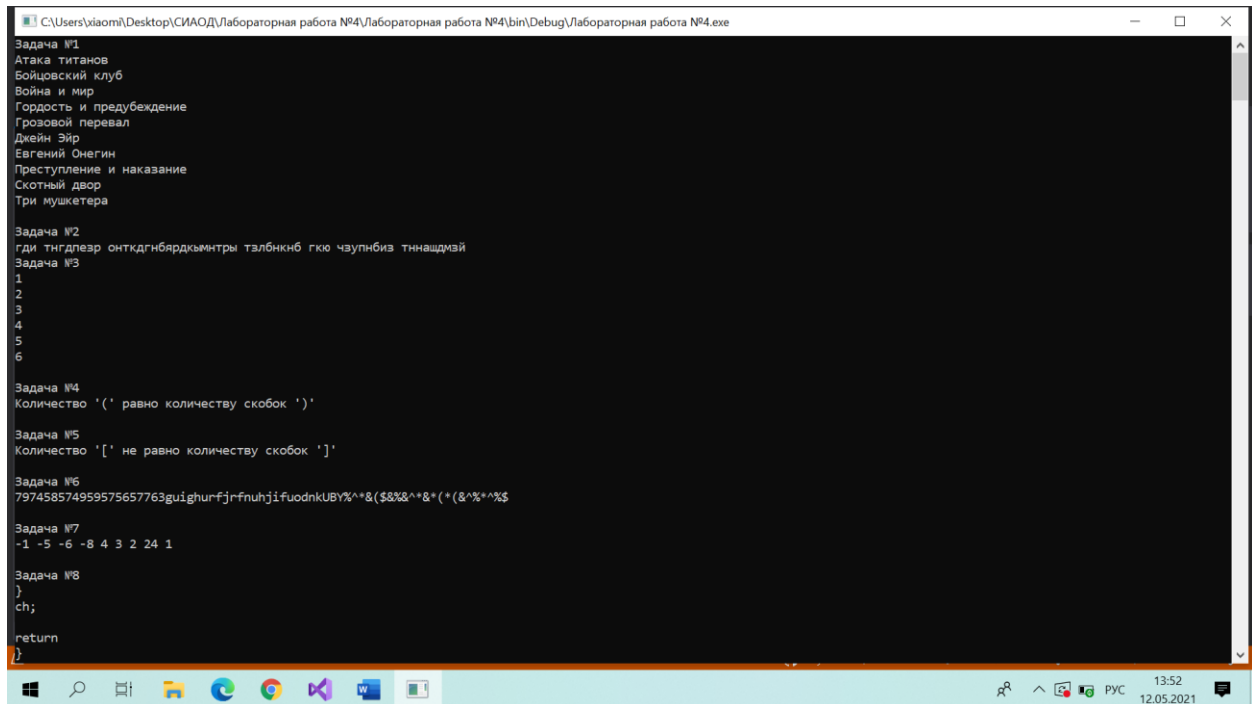
for (int i = String10.Length - 1; i >= 0; i--)
{
    if (String10[i] != ' ') St10.Push(String10[i]);
}
Console.WriteLine(Task10(St10));

Console.WriteLine("\r\n" + "Задача №11" );
String String11 = "";
using (StreamReader sr = new StreamReader("task11.txt"))
{
    String11 = sr.ReadToEnd();
}
MyStack<char> St11 = new MyStack<char>(String11.Length);
if(String11[0]=='(' && String11[String11.Length-1] == ')')
{
    String11 = String11.Substring(1, String11.Length - 2);
}
for(int i=String11.Length-1;i>=0;i--)
{
    if (String11[i] != ' ') St11.Push(String11[i]);
}
Task11(St11);
Console.ReadLine();

Console.ReadLine();
}

```

## Результат работы программы:



```
Задача №1
Атака титанов
Бойцовский клуб
Война и мир
Гордость и предубеждение
Грозовой перевал
Джейн Эйр
Евгений Онегин
Преступление и наказание
Скотный двор
Три мушкетера

Задача №2
гди тнгдлеар онткдгнбярдымнтры тэлбнкнб гюо чзупнбиз тннашдмэй

Задача №3
1
2
3
4
5
6

Задача №4
Количество '(' равно количеству скобок ')'

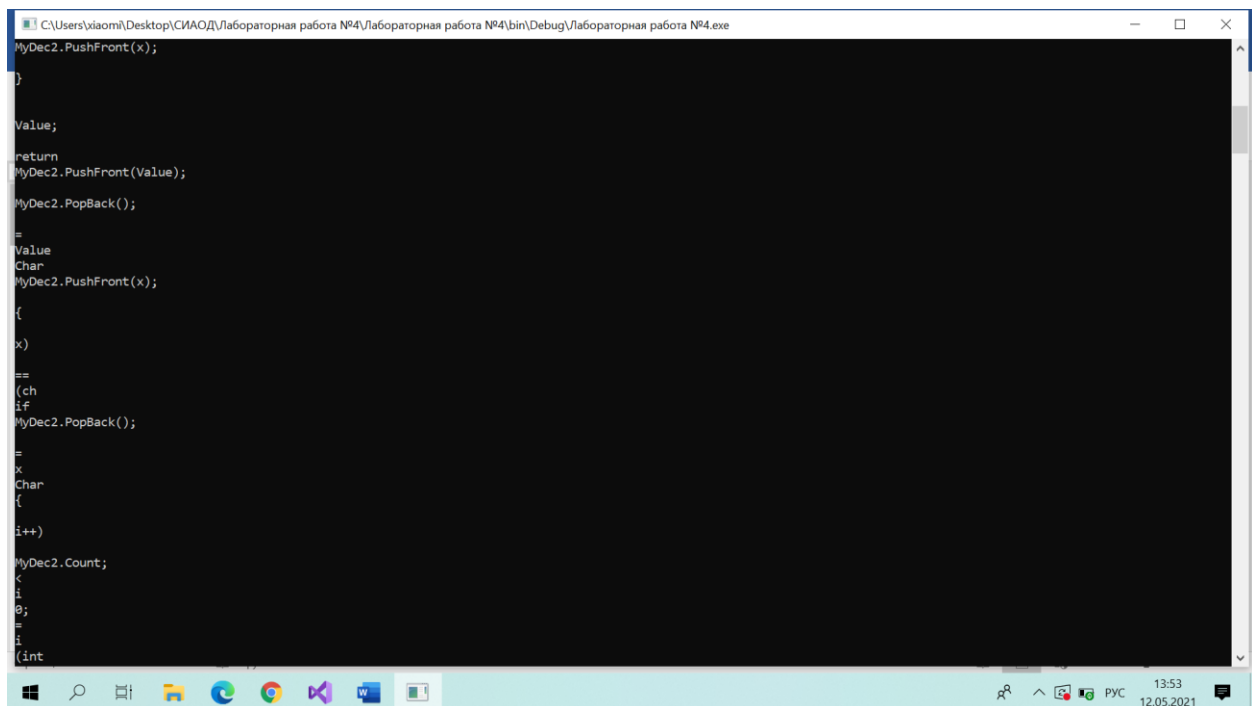
Задача №5
Количество '[' не равно количеству скобок ']'

Задача №6
797458574959575657763guighurfjrfnuhjifuodnkUBV%^^&($$&^^&*(^&%^&^$

Задача №7
-1 -5 -6 -8 4 3 2 24 1

Задача №8
}
ch;
return
}
```

Рисунок 1 – Результат



```
MyDec2.PushFront(x);
}

Value;

return
MyDec2.PushFront(Value);

MyDec2.PopBack();

=
Value
Char
MyDec2.PushFront(x);
{
(x)
==
(ch
if
MyDec2.PopBack();
=
x
Char
{
i++)
MyDec2.Count;
<
i
0;
=
i
(int
```

Рисунок 2 – Результат

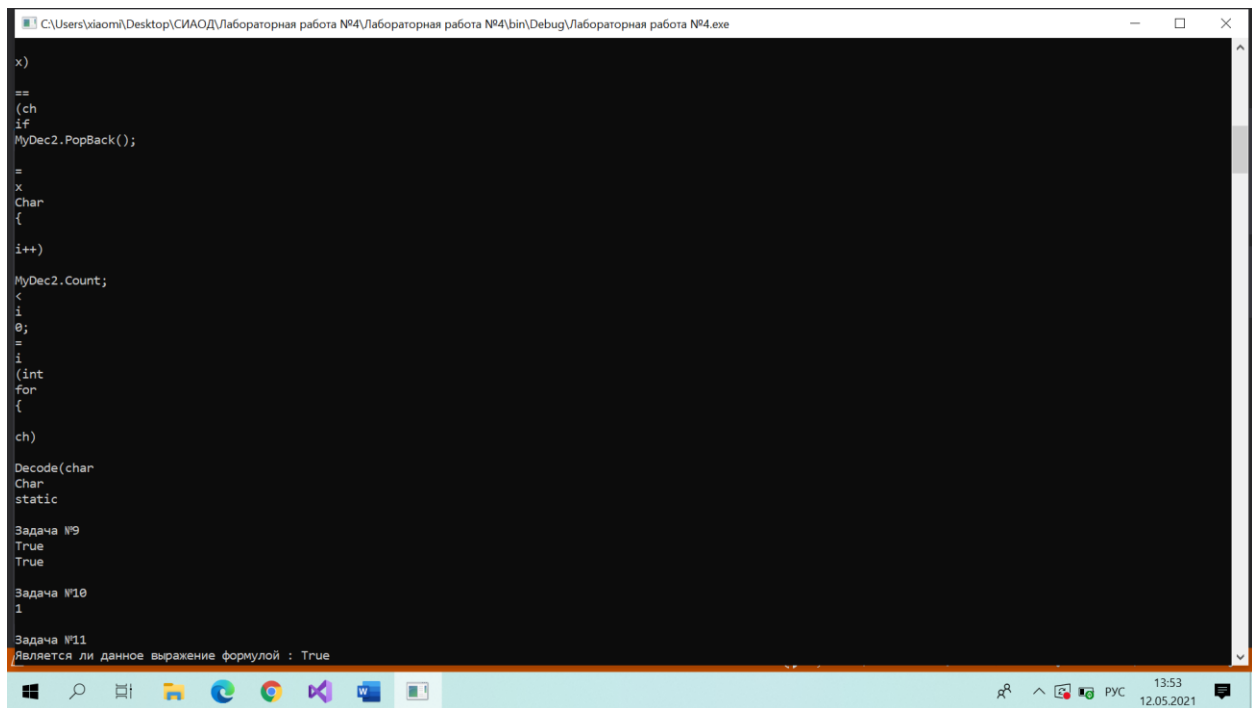


Рисунок 3 – Результат

Вывод: реализовала стек и дек, выполнила все задания.