МИНИСТЕРСТВО ЦИФРОВОГО РАЗВИТИЯ СВЯЗИ И

МАССОВЫХ КОММУНИКАЦИЙ РОССИЙСКОЙ ФЕДЕРАЦИИ

государственное бюджетное образовательное учреждение высшего образование ордена Трудового Красного Знамени

«Московский технический университет связи и информатики»

Кафедра Математической кибернетики и информационных технологий

Отчет по лабораторной работе №4

Реализация стека/дека

по дисциплине «Структуры и алгоритмы обработки данных»

Выполнил: студент группы БФИ1902

Шацкий Е.И

Проверил:

Мкртчян Г. М

Москва 2021 г.

Задания:

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 A на стержень C,
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 B.
Реализовать алгоритм, используя три стека вместо стержней A, B, C. 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:
< ЛВ > ::= T | F | (N<ЛВ>) | (<ЛВ>A<ЛВ>) | (<ЛВ>X<ЛВ>) | (<ЛВ>O<ЛВ>),
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:
<Формула> ::= <Цифра> | M(<Формула>,<Формула>) |
N(Формула>,<Формула>)
< Цифра > ::= 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 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:
< Формула > ::= < Терм > | < Терм > + < Формула > | < Терм > - < Формула >
< Терм > ::= < Имя > | (< Формула >)
< Имя > ::= x | y | z

Ход выполнения лабораторной работы:
Результат выполнения задний 1-11 в консоли показан на рисунке 1 и 2.

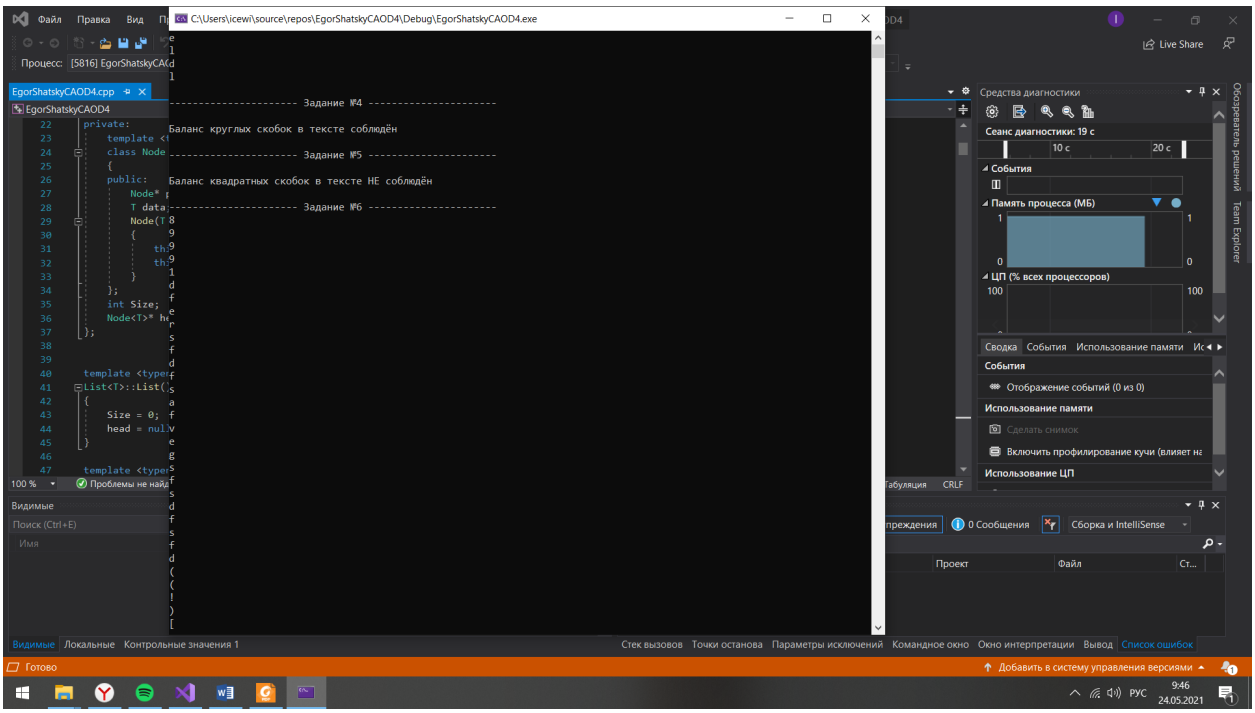

Рисунок 1 – Первая часть вывода в консоль

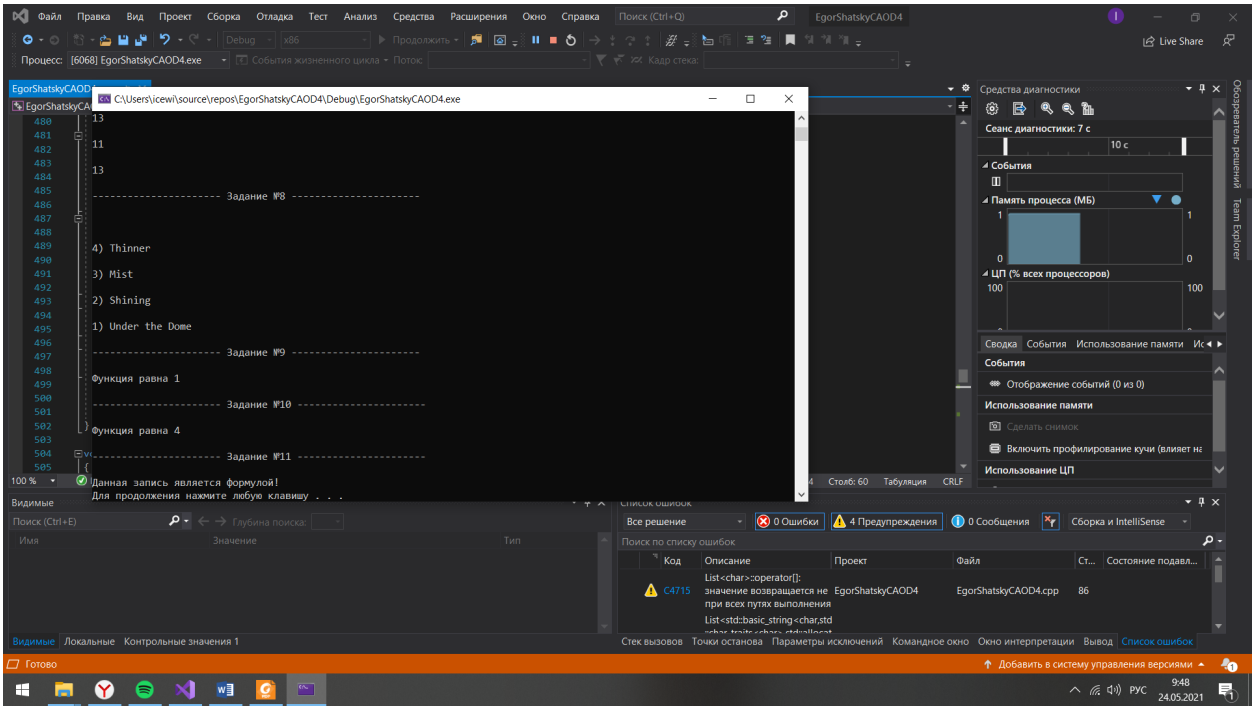

Рисунок 2 - Вторая часть вывода в консоль

Листинг программы

```
#include <iostream>
#include <string>
#include <fstream>
using namespace std;

template <typename T>
class List
{
public:
	List();
	~List();

	void pop_back();
	void pop_front();
	void push_back(T data);
	void push_front(T data);
	void insert(T data, int index);
	void removeAt(int index);
	void clear();
	int GetSize() { return Size; }
	T& operator[](const int index);
private:
	template <typename T>
	class Node
	{
	public:
		Node* pNext;
```

```
            T data;
            Node(T data = T(), Node* pNext = nullptr)
            {
                this->data = data;
                this->pNext = pNext;
            }
        };
        int Size;
        Node<T>* head;
};


template <typename T>
List<T>::List()
{
        Size = 0;
        head = nullptr;
}

template <typename T>
List<T>::~List()
{
        clear();
}

template <typename T>
void List<T>::push_back(T data)
{
```

```
    if (head == nullptr)
    {
        head = new Node<T>(data); // Создаём новый объект типа Node в динамической памяти, передаём в конструктор наши данные, head присваиваем возвращённый указатель на адрес этого объекта в памяти
    }
    else
    {
        Node<T>* current = this->head; // Временному указателю присваиваем значение нашего заголовка head
        while (current->pNext != nullptr) // Идём по порядку по всем адресам, пока pNext не укажет на nullptr, смотрим у текущего элемента поле pNext, указывающее на следующий адрес
        {
            current = current->pNext; // Присваиваем указатель на следующий элемент (присваиваем следующий адрес)
        }
        current->pNext = new Node<T>(data); // Присваиваем адрес нового объекта типа Node pNext, указывающему на nullptr (добавим в поле pNext)
    }
    Size++; // Увеличиваем счётчик количества элементов списка
}


template <typename T>
T& List<T>::operator[](const int index)
{
    int counter = 0; // Счётчик, считающий в каком элементе списка мы сейчас находимся
    Node<T>* current = this->head; // Временному указателю присваиваем значение нашего заголовка head
    while (current != nullptr) // Пока адрес не равен nullptr
```

```
	{
		if (counter == index)
		{
			return current->data;
		}
		current = current->pNext;
		counter++;
	}
}

template <typename T>
void List<T>::pop_front()
{
	Node<T>* temp = head;
	head = head->pNext;
	delete temp;
	Size--;
}

template <typename T>
void List<T>::clear()
{
	while (Size)
	{
		pop_front();
	}
}
```

```
template <typename T>
void List<T>::push_front(T data)
{
	head = new Node<T>(data, head);
	Size++;
}

template <typename T>
void List<T>::insert(T data, int index)
{
	if (index == 0)
	{
		push_front(data);
	}
	else
	{
		Node<T>* previous = this->head;
		for (int i = 0; i < index - 1; i++)
		{
			previous = previous->pNext; // Находим элемент, индекс которого предшествует нужному нам индексу
		}
		Node<T>* newNode = new Node<T>(data, previous->pNext);
		previous->pNext = newNode;
		Size++;
	}
}
```

```
template <typename T>
void List<T>::removeAt(int index)
{
	if (index == 0)
	{
		pop_front();
	}
	else
	{
		Node<T>* previous = this->head;
		for (int i = 0; i < index - 1; i++)
		{
			previous = previous->pNext; // Находим элемент, индекс которого предшествует нужному нам индексу
		}
		Node<T>* toDelete = previous->pNext; // Заносим сюда адрес элемента, который хотим удалить
		previous->pNext = toDelete->pNext;
		delete toDelete;
		Size--;
	}
}

template <typename T>
void List<T>::pop_back()
{
	removeAt(Size - 1);
}
```

```
List<int> lst0;
List<int> lst;
List<string> lst1;
List<char> lst2;
List<bool> lst3;
List<char> lst4;
List<char> lst6;


void Task2() //abcdefghijklmnopqrstuvwxyz dcjgv
{
	lst4.push_back('a'); lst4.push_back('b'); lst4.push_back('c'); lst4.push_back('d'); lst4.push_back('e');
	lst4.push_back('f'); lst4.push_back('g'); lst4.push_back('h'); lst4.push_back('i'); lst4.push_back('j');
	lst4.push_back('k'); lst4.push_back('l'); lst4.push_back('m'); lst4.push_back('n'); lst4.push_back('o');
	lst4.push_back('p'); lst4.push_back('q'); lst4.push_back('r'); lst4.push_back('s'); lst4.push_back('t');
	lst4.push_back('u'); lst4.push_back('v'); lst4.push_back('w'); lst4.push_back('x'); lst4.push_back('y');
	lst4.push_back('z');
	ifstream fin;
	fin.open("Task2.txt");
	if (!fin.is_open())
	{
		cout << endl << "Ошибка в чтении файла, попробуйте еще раз" << endl;
	}
	else
	{
```

```
		cout << endl << "---------------------- Задание №2 ----------------------"
<< endl;
		char ch;
		while (fin.get(ch))
		{
			for (int i = 0; i < lst4.GetSize(); i++)
			{
				if (ch == lst4[i])
				{
					for (int j = 0; j < i; j++)
					{
						lst2.push_back(lst4[0]);
						lst4.pop_front();
					}
					for (int n = 0; n < i; n++)
					{
						lst4.push_back(lst2[0]);
						lst2.pop_front();
					}
					cout << lst4[2] << endl; //Сдвиг на 2
				}
			}
		}
	}
	fin.close();
	lst2.clear();
	lst4.clear();
}
```

```
void Task4() //(((((a++)))find)md)
{
	ifstream fin;
	fin.open("Task4.txt");
	if (!fin.is_open())
	{
		cout << endl << "Ошибка в чтении файла, попробуйте еще раз" << endl;
	}
	else
	{
		cout << endl << "---------------------- Задание №4 ----------------------" << endl;
		char ch;
		while (fin.get(ch))
		{
			if (ch == '(')
			{
				lst2.push_front('(');
			}
			if (ch == ')')
			{
				lst2.pop_front();
			}
		}
		if (lst2.GetSize() == 0)
		{
```

```
                cout << endl << "Баланс круглых скобок в тексте соблюдён"
<< endl;
            }
            else
            {
                cout << endl << "Баланс круглых скобок в тексте НЕ
соблюдён" << endl;
            }
        }
        fin.close();
        lst2.clear();
}

void Task5() //[[[[[a++]]]find]]md]
{
        ifstream fin;
        fin.open("Task5.txt");
        if (!fin.is_open())
        {
            cout << endl << "Ошибка в чтении файла, попробуйте еще раз" <<
endl;
        }
        else
        {
            cout << endl << "---------------------- Задание №5 ----------------------"
<< endl;
            char ch;
            while (fin.get(ch))
            {
```

```
                if (ch == '[')
                {
                        lst2.push_front('[');
                }
                if (ch == ']')
                {
                        lst4.push_back(']');
                }
        }
        if (lst2.GetSize() == lst4.GetSize())
        {
                cout << endl << "Баланс квадратных скобок в тексте
соблюдён" << endl;
        }
        else
        {
                cout << endl << "Баланс квадратных скобок в тексте НЕ
соблюдён" << endl;
        }
    }
    fin.close();
    lst2.clear();
    lst4.clear();
}

void Task6() //dfsdf8s9(fdf8s9afasf!sdf99)[9s]fdf
{
    ifstream fin;
    fin.open("Task6.txt");
```

```
	if (!fin.is_open())
	{
		cout << endl << "Ошибка в чтении файла, попробуйте еще раз" << endl;
	}
	else
	{
		cout << endl << "---------------------- Задание №6 ----------------------" << endl;
		char ch;
		while (fin.get(ch))
		{
			if (isdigit(ch))
			{
				lst2.push_front(ch);
			}
			else if (ch >= 'a' && ch <= 'z')
			{
				lst4.push_front(ch);
			}
			else
			{
				lst6.push_front(ch);
			}
		}
		for (int i = lst2.GetSize() - 1; i > 0; i--)
		{
			cout << lst2[i] << endl;
		}
```

```
		for (int i = lst4.GetSize() - 1; i > 0; i--)
		{
			cout << lst4[i] << endl;
		}
		for (int i = lst6.GetSize() - 1; i > 0; i--)
		{
			cout << lst6[i] << endl;
		}
	}
	fin.close();
	lst2.clear();
	lst4.clear();
	lst6.clear();
}

void Task7() //12 16 19 -27 -33 41 -8 -6
{
	ifstream fin;
	fin.open("Task7.txt");
	if (!fin.is_open())
	{
		cout << endl << "Ошибка в чтении файла, попробуйте еще раз" << endl;
	}
	else
	{
		cout << endl << "---------------------- Задание №7 ----------------------" << endl;
		int ia;
```

```
        string str;
        while (!fin.eof())
        {
            str = "";
            getline(fin, str);
            ia = stoi(str);
            if (ia >= 0)
            {
                lst.push_back(ia);
            }
            if (ia <= 0)
            {
                lst0.push_back(ia);
            }
        }
        for (int i = 0; i < lst.GetSize(); i++)
        {
            lst0.push_back(lst[i]);
        }
        for (int i = 0; i < lst0.GetSize(); i++)
        {
            cout << endl << lst0[i] << endl;
        }
    }
    fin.close();
    lst0.clear();
    lst.clear();
}
```

```
void Task8()
{
	ifstream fin;
	fin.open("Task8.txt");
	if (!fin.is_open())
	{
		cout << endl << "Ошибка в чтении файла, попробуйте еще раз" << endl;
	}
	else
	{
		cout << endl << "---------------------- Задание №8 ----------------------" << endl;
		string str;
		while (!fin.eof())
		{
			str = "";
			getline(fin, str);
			lst1.push_front(str);
		}
		for (int i = 0; i < lst1.GetSize(); i++)
		{
			cout << endl << lst1[i] << endl;
		}
	}
	fin.close();
	lst1.clear();
}
```

```
void Task9() //(N((TOF)A(FOF)))
{
	ifstream fin;
	fin.open("Task9.txt");
	if (!fin.is_open())
	{
		cout << endl << "Ошибка в чтении файла, попробуйте еще раз" << endl;
	}
	else
	{
		cout << endl << "---------------------- Задание №9 ----------------------" << endl;
		bool boo;
		char ch;
		while (fin.get(ch))
		{
			if (ch == 'N' || ch == 'A' || ch == 'X' || ch == 'O')
			{
				lst2.push_front(ch);
			}
			if (ch == 'T' || ch == 'F')
			{
				if (ch == 'T')
					boo = true;
				else
					boo = false;
				lst3.push_front(boo);
```

```
}
if (ch == ')')
{
        if (lst2[0] == 'O')
        {
                if (lst3[0] || lst3[1])
                {
                        boo = true;
                        lst3.pop_front();
                        lst3.pop_front();
                        lst3.push_front(boo);
                }
                else
                {
                        boo = false;
                        lst3.pop_front();
                        lst3.pop_front();
                        lst3.push_front(boo);
                }
        }
        if (lst2[0] == 'A')
        {
                if (lst3[0] && lst3[1])
                {
                        boo = true;
                        lst3.pop_front();
                        lst3.pop_front();
                        lst3.push_front(boo);
```

```
            }
            else
            {
                boo = false;
                lst3.pop_front();
                lst3.pop_front();
                lst3.push_front(boo);
            }
        }
        if (lst2[0] == 'X')
        {
            if (lst3[0] || lst3[1])
            {
                boo = true;
                lst3.pop_front();
                lst3.pop_front();
                lst3.push_front(boo);
            }
            else
            {
                boo = false;
                lst3.pop_front();
                lst3.pop_front();
                lst3.push_front(boo);
            }
        }
        if (lst2[0] == 'N')
        {
```

```
                                        if (!lst3[0])
                                        {
                                                boo = true;
                                                lst3.pop_front();
                                                lst3.push_front(boo);
                                        }
                                        else
                                        {
                                                boo = false;
                                                lst3.pop_front();
                                                lst3.push_front(boo);
                                        }
                                }
                                lst2.pop_front();
                        }
                }
                cout << endl << "Функция равна " << lst3[0] << endl;
        }
        fin.close();
        lst3.clear();
        lst2.clear();
}

void Task10() //M(8,N(9,4))
{
        ifstream fin;
        fin.open("Task10.txt");
        if (!fin.is_open())
```

```
	{
		cout << endl << "Ошибка в чтении файла, попробуйте еще раз" << endl;
	}
	else
	{
		cout << endl << "---------- Задание №10 ----------" << endl;
		int ia;
		char ch;
		while (fin.get(ch))
		{
			if (ch == 'M' || ch == 'N')
			{
				lst2.push_front(ch);
			}
			if (isdigit(ch))
			{
				ia = ch - '0'; // Приведение типов
				lst.push_front(ia);
			}
			if (ch == ')')
			{
				if (lst2[0] == 'N')
				{
					if (lst[0] > lst[1])
						lst.push_front(lst[1]);
					if (lst[0] < lst[1])
						lst.push_front(lst[0]);
```

```
                        lst.pop_front();
                        lst.pop_front();
                    }
                    if (lst2[0] == 'M')
                    {
                        if (lst[0] > lst[1])
                            lst.push_front(lst[0]);
                        if (lst[0] < lst[1])
                            lst.push_front(lst[1]);
                        lst.pop_front();
                        lst.pop_front();
                    }
                    lst2.pop_front();
                }

            }
            cout << endl << "Функция равна " << lst[0] << endl;
        }
        fin.close();
        lst.clear();
        lst2.clear();
}

void Task11() //x+y-z+z
{
        ifstream fin;
        fin.open("Task11.txt");
        if (!fin.is_open())
```

```
	{
		cout << endl << "Ошибка в чтении файла, попробуйте еще раз" << endl;
	}
	else
	{
		cout << endl << "---------- Задание №11 ----------" << endl;
		char ch;
		while (fin.get(ch))
		{
			if (ch == 'x' || ch == 'y' || ch == 'z') //Операнды
			{
				lst2.push_front(ch);
			}
			if (ch == '+' || ch == '-') //Оператор
			{
				lst4.push_front(ch);
			}
			if (lst2.GetSize() == 3 && lst4.GetSize() == 0)
			{
				cout << endl << "Данная запись НЕ является формулой!" << endl;
				break;
			}
			if (lst2.GetSize() == 0 && lst4.GetSize() == 3)
			{
				cout << endl << "Данная запись НЕ является формулой!" << endl;
```

```
                        break;
                    }
                    if (lst2.GetSize() == 1 && lst4.GetSize() == 2)
                    {
                        cout << endl << "Данная запись НЕ является
формулой!" << endl;
                        break;
                    }
                    if (lst2.GetSize() == 2 && lst4.GetSize() == 1) //Может быть и
польская нотация
                    {
                        lst2.pop_front();
                        lst4.pop_front();
                    }
                }
                if (lst4.GetSize() == 0 && lst2.GetSize() == 1)
                {
                    cout << endl << "Данная запись является формулой!" <<
endl;
                }
                else
                {
                    cout << endl << "Данная запись НЕ является формулой!" <<
endl;
                }
            }
            fin.close();
            lst4.clear();
            lst2.clear();
```

```
}

int main()
{
	setlocale(LC_ALL, "ru");
	//Task1();
	Task2();
	//Task3();
	Task4();
	Task5();
	Task6();
	Task7();
	Task8();
	Task9();
	Task10();
	Task11();
	system("pause");
	return 0;
}
```

Вывод:
В данной лабораторной были реализованы стек,дек и произведена работа с ними на языке C++.