**ФГБОУ ВО**

**Национальный исследовательский университет**

**«МЭИ»**

**Лабораторная работа №2**

**Динамические структуры данных**

**Вариант 8**

**Задание №2:**

**Списки**

**Выполнил:**

студент группы Аэ-21-23

Нестеров А.С.

**Москва, 2024**

1. **Постановка задачи**

Задание:

список однонаправленный целых чисел

Формирование списка из n элементов

Вставка элемента после элемента со значением X

Удаление первого элемента

проверить, есть ли в списке L хотя бы два одинаковых элемента

Функции программы:

1. Формирование списка из n элементов
2. Вставка элемента после элемента со значением X
3. Удаление первого элемента
4. проверить, есть ли в списке L хотя бы два одинаковых элемента

Входные данные:

* n(int) – количество элементов списка [1;10]
* Элементы списка (int) – ограничение (4 байта)

[-2147483648; 2147483647]

1. **Разработка приложения** 
   1. Разработка структуры приложения

Задание реализовано в виде консольного приложения на языке С++ с использованием процедур и функций.

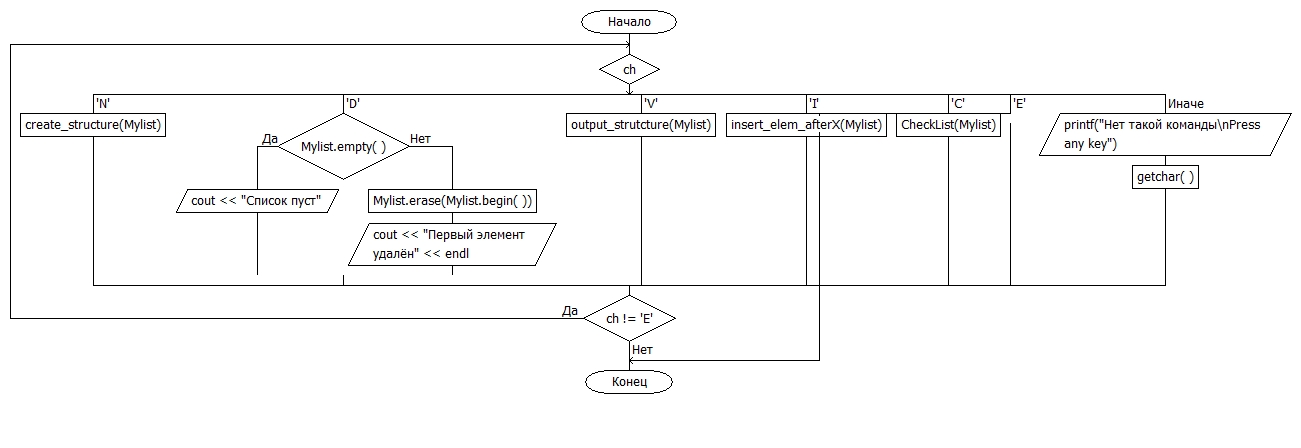


Рисунок 1 – Структура приложения

Таблица 1 - Спецификация функций

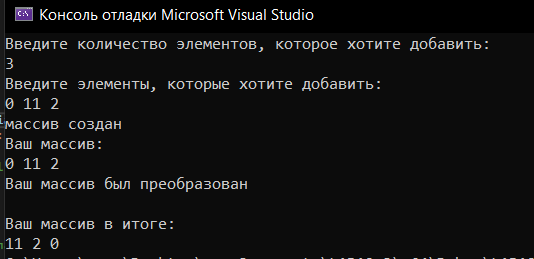
|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| № | Название | Назначение | Вх. данные | Вых. данные |
| 1 | create\_structure | создание списка из n элементов | Список  (количество элементов и сами элементы функция запрашивает сама) | список |
| 2 | Mylist.erase(Mylist.begin()); | Удалить первый элемент | Список | Список |
| 3 | insert\_elem\_afterX(Mylist); | Вставка элемента после элемента со значением X | Список | Список |
| 4 | CheckList(Mylist); | проверить, есть ли в списке L хотя бы два одинаковых элемента | Список | Список |
|  |  |  |  |  |

2.3 Разработка пользовательского интерфейса

Категория пользователей: Студенты

Интерфейс представляет собой консольное приложение, в котором взаимодействие пользователя с программой осуществляется с помощью диалогового окна с выбором опций.

Пример диалогов с пользователем:



1. **Реализация и тестирование приложения**

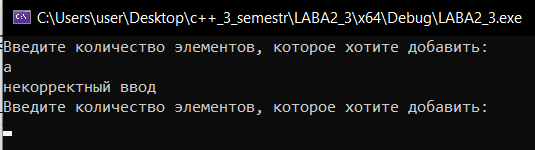
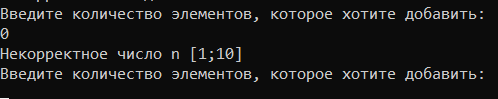
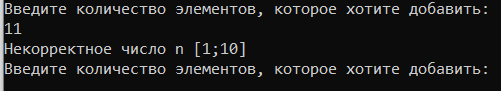
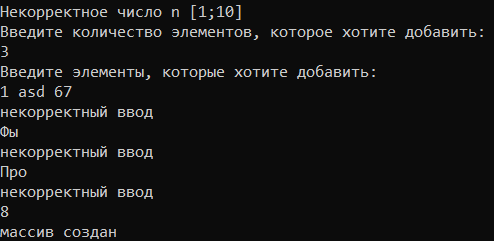
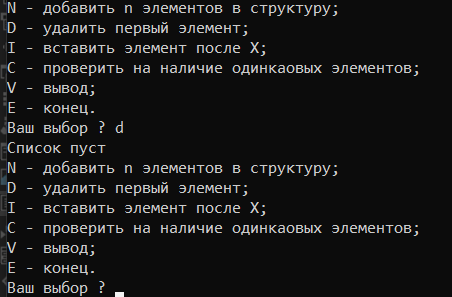
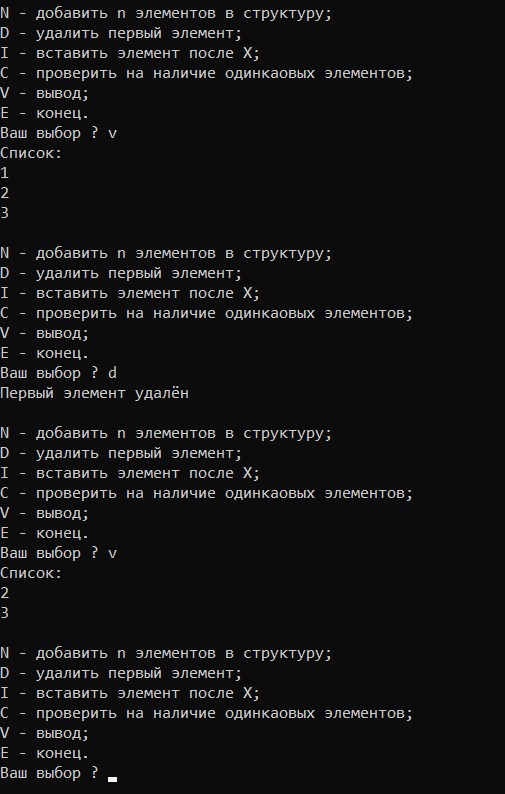
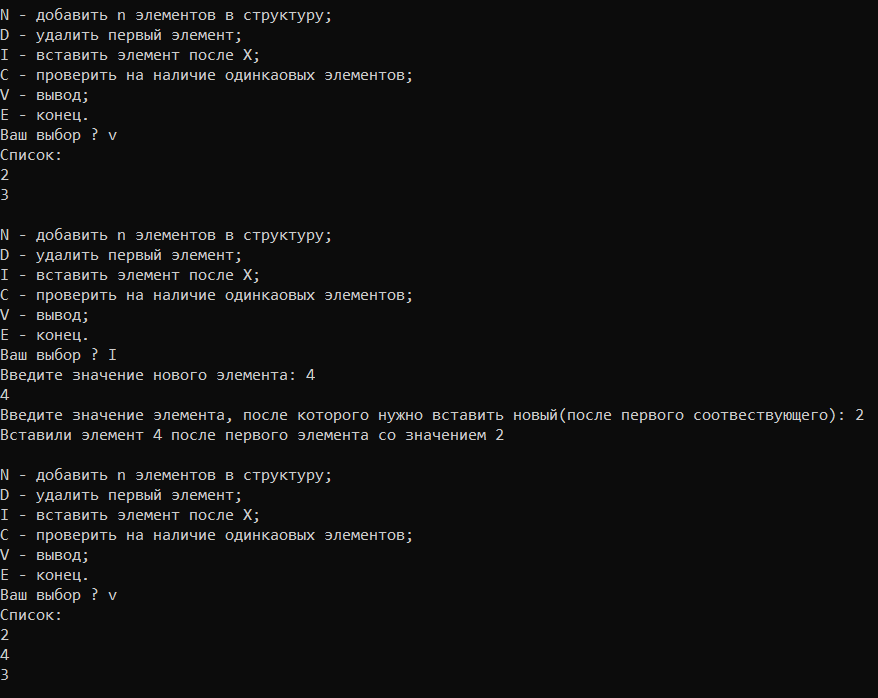
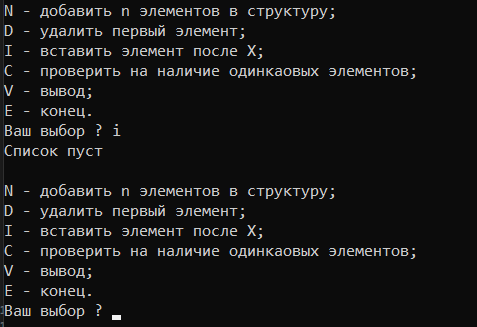
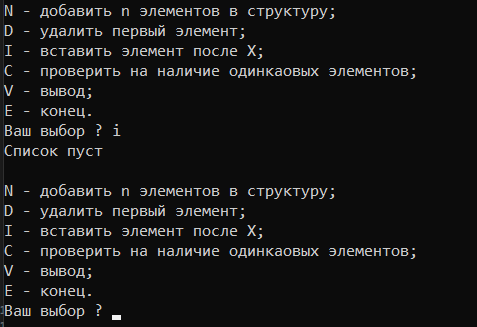
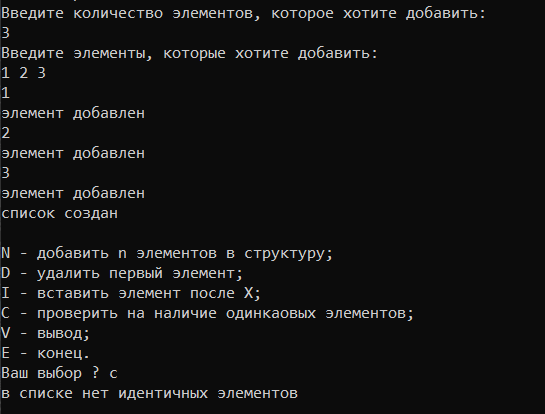
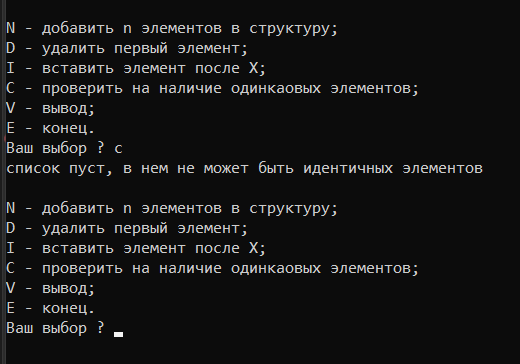
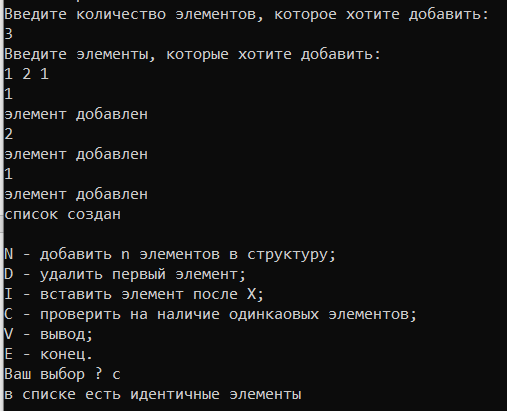
3.1 Описание разработанной программы

Данная программа позволяет пользователю создать собственную очередь, содержащую элементы типа char, добавить элементы в стек/очередь, очистить стек, просмотреть стек/очередь, преобразовать стек в очередь и завершить работу программы в режиме диалога.

3.2 Тестирование программы

Таблица 2 – Тесты программы

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| № | Тестируемая функция | Входные данные | Ожидаемый результат | Цель теста |
| 1 | create\_structure | N = a | некорректный ввод  Введите количество элементов, которое хотите добавить: | Некорректное число n |
| 2 | create\_structure | N=0 | Некорректное число n [1;10]  Введите количество элементов, которое хотите добавить: | Некорректное число n |
| 3 | create\_structure | N = 11 | Некорректное число n [1;10]  Введите количество элементов, которое хотите добавить: | Некорректное число n |
| 4 | create\_structure | N = 3  1 asd 67  Фы  Про  8 | Введите элементы, которые хотите добавить:  1 asd 67  некорректный ввод (запрос ввести заново)  фы  некорректный ввод  (запрос ввести заново)  про  некорректный ввод  (запрос ввести заново)  8  массив создан  Ваш массив:  1 67 8 | Некорректный элемент |
| 5 | Mylist.erase(Mylist.begin()); | Пустой список | Список пуст | Из пустого списка невозможно удалить элемент |
| 6 | Mylist.erase(Mylist.begin()); | 1 2 3 | 2 3 | Проверка на удаление |
| 7 | insert\_elem\_afterX(Mylist); | 2 3  Запрашивается внутри:  Newelem = 4  X = 2 | 2 4 3 | Проверка работы функции |
| 8 | insert\_elem\_afterX(Mylist); | Пустой список | Список пуст | В пустой список невозможно вставить элемент после какого-то элемента |
| 9 | insert\_elem\_afterX(Mylist); | N = 3  1 2 3  Newelem = 12  X = 4 | В списке нет такого элемента | В список невозможно вставить newelem |
| 10 | CheckList(Mylist); | Список без идентичных элементов | Список без идентичных элементов | Список без идентичных элементов |
| 11 | CheckList(Mylist); | Пустой список | Пустой список | Пустой список |
| 12 | CheckList(Mylist); | 1 2 1 | Есть идентичные элементы | Есть идентичные элементы |

1. 
2. 
3. 
4. 
5. 
6. 
7. 
8. 
9. 
10. 
11. 
12. 

1. **Код**