МИНИСТЕРСТВО НАУКИ И ВЫСШЕГООБРАЗОВАНИЯ

РОССИЙСКОЙ ФЕДЕРАЦИИ

Федеральное государственное автономное образовательное учреждение высшего образования   
**«Национальный исследовательский   
Нижегородский государственный университет им. Н.И. Лобачевского»**

**(ННГУ)**

**Институт информационных технологий, математики и механики**

**ЛАБОРАТОРНАЯ РАБОТА**

на тему:

**«Стек. Постфиксная форма.»**

**Выполнил(а):** студент группы 3822Б1ФИ2

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ / Савченко М.П./

Подпись

**Проверил:** к.т.н, доцент каф. ВВиСП

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ / Кустикова В.Д./

Подпись

Нижний Новгород  
2023

**Содержание**

[Введение 3](#_Toc147915966)

[1 Постановка задачи 4](#_Toc147915967)

[2 Руководство пользователя 5](#_Toc147915968)

[2.1 Приложение для демонстрации работы битовых полей 5](#_Toc147915969)

[2.2 Приложение для демонстрации работы множеств 5](#_Toc147915970)

[2.3 «Решето Эратосфено» 5](#_Toc147915971)

[3 Руководство программиста 6](#_Toc147915972)

[3.1 Описание алгоритмов 6](#_Toc147915973)

[3.1.1 Битовые поля 6](#_Toc147915974)

[3.1.2 Множества 6](#_Toc147915975)

[3.1.3 «Решето Эратосфена» 6](#_Toc147915976)

[3.2 Описание программной реализации 6](#_Toc147915977)

[3.2.1 Описание класса TBitField 6](#_Toc147915978)

[3.2.2 Описание класса TSet 7](#_Toc147915979)

[Заключение 8](#_Toc147915980)

[Литература 9](#_Toc147915981)

[Приложения 10](#_Toc147915982)

[Приложение А. Реализация класса TBitField 10](#_Toc147915983)

[Приложение Б. Реализация класса TSet 10](#_Toc147915984)

# Введение

Стек и постфиксная (или обратная польская) форма арифметического выражения - это концепции, которые могут быть актуальными и полезными в различных областях программирования и вычислительной математики.

1. **Стек (Stack):**

Стек - это структура данных, работающая по принципу "последний вошел, первый вышел" (Last In, First Out - LIFO). Это означает, что элементы добавляются и удаляются только с одного конца стека (вершины). Стек может быть реализован как массив или связанный список.

Применение стека:

1. **Управление вызовами функций:** Стек используется для хранения информации о вызовах функций, чтобы знать, куда возвращаться после завершения каждой функции.
2. **Обратная трассировка (debugging):** Стек помогает отслеживать порядок вызовов функций и точки, в которых произошла ошибка.
3. **Вычисления с использованием рекурсии:** Рекурсивные алгоритмы часто используют стек для хранения промежуточных результатов.
4. **Постфиксная форма:**

Постфиксная форма (или обратная польская запись) - это способ записи арифметических выражений, при котором операторы расположены после своих операндов. Это исключает необходимость в скобках и упрощает вычисление выражений.

Применение постфиксной формы:

1. **Калькуляторы:** Некоторые карманные калькуляторы используют постфиксную форму для упрощения вычислений.
2. **Оптимизация вычислений:** Постфиксная форма позволяет избежать проблем с приоритетом операторов и порядком операций, делая выражения более однозначными.
3. **Автоматическая генерация кода:** Некоторые компиляторы используют постфиксную форму внутри своих промежуточных представлений.

Вместе стек и постфиксная форма могут использоваться, например, для вычисления постфиксных выражений без использования рекурсии. Выражение обрабатывается слева направо, операнды помещаются в стек, и когда встречается оператор, извлекаются нужное количество операндов из стека, выполняется операция, и результат помещается обратно в стек.

Хотя стек и постфиксная форма могут казаться несколько абстрактными, они оказываются полезными инструментами в различных областях программирования и алгоритмов, особенно в тех случаях, когда нужно эффективно управлять порядком операций и сохранять контекст выполнения.

# Постановка задачи

Цель –

Задачи:

…

# Руководство пользователя

## Приложение для демонстрации работы битовых полей

1. Запустите приложение с названием \*.exe. В результате появится окно, показанное ниже (рис. 1).



1. Основное окно программы

## Приложение для демонстрации работы множеств

## «Решето Эратосфено»

# Руководство программиста

## Описание алгоритмов

### Битовые поля

### Множества

### «Решето Эратосфена»

## Описание программной реализации

### Описание класса TBitField

class TBitField

{

private:

int BitLen; // длина битового поля - макс. к-во битов

TELEM \*pMem; // память для представления битового поля

int MemLen; // к-во эл-тов Мем для представления бит.поля

// методы реализации

int GetMemIndex(const int n) const;

TELEM GetMemMask (const int n) const;

public:

TBitField(int len);

TBitField(const TBitField &bf);

…

};

Назначение: представление битового поля.

Поля:

BitLen – длина битового поля – максимальное количество битов.

pMem – память для представления битового поля.

MemLen – количество элементов для представления битового поля.

Методы:

int GetMemIndex(const int n) const;

Назначение: получение индекса элемента в памяти…

Входные параметры:

n – номер бита.

Выходные параметры:

Номер элемента в памяти.

### Описание класса TSet

# Заключение

Что сделано?

# Литература

1. Название страницы [ссылка].
2. Фамилия И.О. Название. – Издательство, год. – количество страниц с
3. .

# Приложения

## Приложение А. Реализация класса TBitField

## Приложение Б. Реализация класса TSet