Министерство науки и высшего образования Российской Федерации

Федеральное государственное бюджетное   
образовательное учреждение высшего образования

«Новгородский государственный университет имени Ярослава Мудрого»

ПОЛИТЕХНИЧЕСКИЙ ИНСТИТУТ

КАФЕДРА ИНФОРМАЦИОННЫХ ТЕХНОЛОГИЙ И СИСТЕМ

Сравнение рекурсивных, циклических алгоритмов

и Проверка слов на палиндромность

Пояснительная записка к лабораторной работе по учебной дисциплине «Основы программирования: Алгоритмические языки и программирование»

по специальности 09.03.01 Информатика и вычислительная техника  
ПТИ.ЛР 4093 004.002ПЗ

|  |  |
| --- | --- |
|  | Руководитель  / И. Ю. Кулаков  «\_\_\_»\_\_\_\_\_\_\_\_\_\_\_\_2025 г. |
|  | Студент группы 4093  / Д.В.Баранов  «\_\_\_»\_\_\_\_\_\_\_\_\_\_\_\_2025 г. |

[Введение 3](#_Toc192029247)

[1 Постановка задачи 4](#_Toc192029248)

[1.1 Наименование задачи 4](#_Toc192029249)

[1.2 Словесное описание 4](#_Toc192029250)

[1.2.1 Ввод 4](#_Toc192029251)

[1.2.2 Вывод 4](#_Toc192029252)

[1.3 Внешние спецификации данных 4](#_Toc192029253)

[1.4 Внешние спецификации функции данных 5](#_Toc192029254)

[1.5 Математическая формулировка задачи 5](#_Toc192029255)

[1.6 Специфика интерфейса 6](#_Toc192029256)

[1.7 Внешние данные тестирования 6](#_Toc192029257)

[1.8 Пример работы программы 7](#_Toc192029258)

[2 Проектирование задачи 8](#_Toc192029259)

[2.1 Наименование программы 8](#_Toc192029260)

[2.3 Выбор метода решения поставленной задачи 9](#_Toc192029261)

[2.4 Уточнение глобальных данных программы и пользовательские типы 9](#_Toc192029262)

[2.5 Декомпозиция функции 13](#_Toc192029263)

[2.6 Алгоритмизация 13](#_Toc192029264)

[Приложение А 17](#_Toc192029265)

# Введение

Задачей лабораторной работы является разработка приложения для анализа эффективности рекурсивных и циклических решений для вычисления математически функций, а также сравнения слова на палиндром. Программа позволяет пользователю вводить число, замерять время выполнения и количество вызовов рекурсии. Это поможет оценить преимущество и недостатки рекурсивного и циклического алгоритма и выбрать наиболее эффективный вариант в зависимости от условий.

Разработка программы может использоваться в различных сферах, где важно оценивать производительность алгоритмов, например в образовании - для обучения студентов принципам рекурсии, в анализе алгоритмов – для тестирования производительности разных методов вычисления.

Задачи:

- Анализ существующих методов решения задачи и их сравнение;

- Разработка рекурсивного и циклического алгоритмов для вычисления функций;

- Реализация системы замера времени выполнения программы;

- Внедрение механизма подсчёта вызовов функций;

- Выполнение проверки слова на палиндром;

- Создание графического интерфейса для удобного взаимодействия с программой;

- Проведение тестирования программы и анализ эффективности работоспособности программы;

- Подготовка документации.

# 1 Постановка задачи

# 1.1 Наименование задачи

Разработка программного приложения для сравнения эффективности рекурсивного и циклического вычисления математических функций, а также проверки слов на палиндромность.

# 1.2 Словесное описание

Программа позволяет пользователю анализировать эффективность рекурсивного, циклического решения математический функций и сравнения слов на палиндромность поочерёдно сравнивая конец и начало слово.

# 1.2.1 Ввод

Пользователя просят ввести число для вычисления функций или слово для проверки на палиндром.

# 1.2.2 Вывод

- Значение функций, вычисляемые двумя способами;

- Время выполнения рекурсивного и циклического решения;

- Количество вызовов функций;

- Результат проверки слова на палиндромность(YES или NO).

# 1.3 Внешние спецификации данных

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Объект программы | Имя этого объекта в программе | Тип данных | Диапазоны представления | Простая/  структура | Вход/  выход/  Константа |
| Source.cpp | n | int | n>=1 | Простая | Ввод |
| Объект программы | Имя этого объекта в программе | Тип данных | Диапазоны представления | Простая/  структура | Вход/  выход/  Константа |
| Source.cpp | resRec1 | long long | -9\*10^20<resRec1<9\*10^20 | Простая | Выход |
| Source.cpp | resCik1 | long long | -9\*10^20<resCic1<9\*10^20 | Простая | Выход |
| Source.cpp | resRec2 | long long | -9\*10^20<resRec2<9\*10^20 | Простая | Выход |
| Source.cpp | resCik2 | long long | -9\*10^20<resCic2<9\*10^20 | Простая | Выход |
| Source.cpp | timeRec | long double | timeRec>0 | Простая | Выход |
| Source.cpp | timeCik | long double | timeCik>0 | Простая | Выход |
| Source.cpp | count | int | count>=0 | Простая | Выход |
| Source.cpp | slovo | string | Слово без пробелов | Структура | Ввод |
| Source.cpp | resPol | bool | true или false | Простая | Вывод |

# 1.4 Внешние спецификации функции данных

- Функция вычисления первой математической функции рекурсивным способом;

- Функция вычисления первой математической функции циклическим способом;

- Функция вычисления второй математической функции рекурсивным способом;

- Функция вычисления второй математической функции циклическим способом;

- Функция проверки слова на палиндром;

- Графический интерфейс для взаимодействия с пользователем.

# 1.5 Математическая формулировка задачи

Математическая модель 1 формулы:

- F(n) = 1, при n <=1;

- F(n) = 3 + F(n - 1) \* F(n - 2) - F(n - 1) - F(n - 2), если n > 1 и n нечётно;

- F(n) = 2 \* F(n - 1), если n > 1 и n чётно.

Математическая модель 2 формулы:

- F(n) = 1, при n = 1;

- F(n) = 2, при n = 2;

- F(n) = [(7\*n + F(n - 3))/9], если n > 2 и n чётно;

- F(n) = [(5\*n + F(n - 1) + F(n - 2))/7], если n > 2 и n нечётно.

Проверка палиндрома:

Функция должна сравнивать первый и последний символ строки, затем вызывать сама себя для оставшейся подстроки.

# 1.6 Специфика интерфейса

Программа имеет графический интерфейс, содержащий:

- Поле ввода для числа, которое будет использоваться в программе;

- Кнопки “Вычисления 1 функции”;

- Кнопки “Вычисления 2 функции”;

- Поле для ввода строки;

- Кнопка “Палиндром”;

- Поле, которое отображает результаты, а точнее значение самой функции, время работы, количество вызовов рекурсии, YES если слово – палиндром, в ином случае NO;

- Меню с отображением формулы функции для удобства пользователя.

# 1.7 Внешние данные тестирования

Возможными ошибками являются введение пользователем отрицательных чисел, некорректный формат по типу букв, символов и переполнение при большом введение числа. Также эти ошибки обрабатываются соответственно: выводится сообщение об ошибке при некорректном введение числа или его формате, при переполнение использовать ограничение на максимальное введение числа.

# 1.8 Пример работы программы

Для проверки работоспособности программы используем тест кейс:

|  |  |  |  |
| --- | --- | --- | --- |
| Test Case ID/Priority | TC001 | | 1 |
| IDEA: Проверка работоспособности программы | | | |
| Revision History | | | |
| Created on 21/02/2025 by Даниил Баранов | | Новый тест кейс | |
| ACTION | | | |
| 1) открыть программу exe.exe;  2) выбрать “Ввести число для функции” ;  2) выбрать 1 функцию ;  3) выбрать способ (рекурсивный);  4) внести число;  6) выбрать “Ввести число для функции”;  7) выбрать 2 функцию ;  8) выбрать способ (циклический);  9) внести число;  11) выбрать “Внести число для проверки”;  12) внести строку; | | | |

Тестовые случаи:

|  |  |  |
| --- | --- | --- |
| Test Case ID | Вводные данные | Результат |
| ТС01 | n = 7 | Вычисление 2 функций, замер времени, количество вызовов |
| ТС02 | “level” | Вывод “YES” (палиндром) |
| ТС03 | “hello” | Вывод “NO” (не палиндром) |
| ТС03 | x = -1 | Сообщение об ошибке |
| Test Case ID | Вводные данные | Результат |
| ТС04 | “1234” | Сообщение об ошибке |

# 2 Проектирование задачи

# 2.1 Наименование программы

Название программы: “Анализ эффективности рекурсивных и циклических алгоритмов вычисляемых по определённым математическим функциям с дополнительной проверкой слов на палиндромность рекурсивным методом”.

2.2 Уточнение словесного описания

Программа предназначена для сравнения эффективности рекурсивной и циклических алгоритмов при вычислении математических функций, а также для проверки слов на палиндромность. Пользователь вводит число или строку в зависимости, что хочет выбрать пользователь, программа выполняет вычисления выбранного алгоритма и выводит результат, а также данные о самом алгоритме.

Для рекурсивного и циклического алгоритма:

- Изначальное число, которое вводит пользователь;

- Математическая формула;

- Результат выполнения;

- Время выполнения функции;

- Количество вызовов.

Для проверки на палиндромность:

- Изначальная строка, которую вводит пользователь;

- Слово “YES” или “NO” в зависимости от проверки.

Программа позволяет студентам на практике изучить различия между рекурсией и итерацией, их слабые и сильные стороны. Кроме этого, она может применяться в исследованиях, связанных с анализом сложности алгоритмов. Проверка слова на палиндромность – это важный элемент работы со строками, что делает программу универсальным инструментом для учебных целей и практического использования в программировании.

# 2.3 Выбор метода решения поставленной задачи

Для того, чтобы вычислить математические функции используются рекурсивный и циклический метод. Рекурсивный метод позволяет строить решение в виде последовательного вызова самой функции, но многоразовый вызов функции требует больше времени из-за копирования паромеров в стек, а также чрезмерное использование памяти, копированием переменных и параметров. Циклический метод, напротив, выполняет вычисления внутри одного блока кода, что делает его более эффективным по потреблению памяти.

Для проверки слова на палиндром применяется метод поочерёдного сравнения символов с начала и конца строки. Этот метод писался также рекурсивным способом, но его хватает для обработки даже длинных строк.

Для замера времени используется стандартная библиотека “chrono”, что позволяет точно измерить разницу во времени между началом и завершением выполнения определённого блока кода.

# 2.4 Уточнение глобальных данных программы и пользовательские типы

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Объект программы | Имя этого объекта в программе | Характеристики | | | Использование в программе |
| Тип данных | Диапазоны представления | Простая/  Структура/  Константа |
| Source.cpp | n1 | int | n1>=1 | Простая | Ввод |
| Объект программы | Имя этого объекта в программе | Характеристики | | | Использование в программе |
| Тип данных | Диапазоны представления | Простая/  Структура/  Константа |
| Source.cpp | res1 | long long | -9\*10^20<res1<9\*10^20 | Простая | Выход |
| Source.cpp | timeRec1 | long double | timeRec1>0 | Простая | Выход |
| Source.cpp | count1 | int | count1>=0 | Простая | Выход |
| Source.cpp | n2 | int | n2>=1 | Простая | Ввод |
| Source.cpp | res2 | long long | -9\*10^20<res2<9\*10^20 | Простая | Выход |
| Source.cpp | timeCic1 | long double | timeCic1>0 | Простая | Выход |
| Source.cpp | count2 | int | count2>=0 | Простая | Выход |
| Source.cpp | n3 | int | n3>=1 | Простая | Ввод |
| Source.cpp | res3 | long long | -9\*10^20<res3<9\*10^20 | Простая | Выход |
| Source.cpp | timeRec2 | long double | timeRec2>0 | Простая | Выход |
| Source.cpp | count3 | int | count3>=0 | Простая | Выход |
| Source.cpp | n4 | int | n4>=1 | Простая | Ввод |
| Source.cpp | res4 | long long | -9\*10^20<res4<9\*10^20 | Простая | Выход |
| Source.cpp | timeCic2 | long double | timeCic2>0 | Простая | Выход |
| Source.cpp | count4 | int | count4>=0 | Простая | Выход |
| Source.cpp | slovo | string | Строка без пробелов | Простая | Ввод |
| Объект программы | Имя этого объекта в программе | Характеристики | | | Использование в программе |
| Тип данных | Диапазоны представления | Простая/  Структура/  Константа |
| Source.cpp | nach | int | nach>=0 | Простая | Индекс символов |
| Source.cpp | kon | int | kon>=0 | Простая | Индекс символов |
| MyForm.h | nRec1 | int | 0<nRec1<=2500 | Простая | Ввод  Вывод |
| MyForm.h | res1 | long long | -9\*10^20<res1<9\*10^20 | Простая | Выход |
| MyForm.h | time1 | long double | time1>0 | Простая | Выход |
| MyForm.h | countRec1 | int | countRec1>=0 | Простая | Выход |
| MyForm.h | nCic1 | int | 0<nCic1<=21\*10^8 | Простая | Ввод  Вывод |
| MyForm.h | res2 | long long | -9\*10^20<res2<9\*10^20 | Простая | Выход |
| MyForm.h | time2 | long double | time2>0 | Простая | Выход |
| MyForm.h | countCic1 | int | countCic1>=0 | Простая | Выход |
| MyForm.h | nRec2 | int | 0<nRec2<=5000 | Простая | Ввод  Вывод |
| MyForm.h | res3 | long long | -9\*10^20<res3<9\*10^20 | Простая | Выход |
| MyForm.h | time3 | long double | time3>0 | Простая | Выход |
| Объект программы | Имя этого объекта в программе | Характеристики | | | Использование в программе |
| Тип данных | Диапазоны представления | Простая/  Структура/  Константа |
| MyForm.h | countRec2 | int | countRec2>=0 | Простая | Выход |
| MyForm.h | nCic2 | int | 0<nCic2<=21\*10^8 | Простая | Ввод  Выход |
| MyForm.h | res4 | long long | -9\*10^20<res4<9\*10^20 | Простая | Выход |
| MyForm.h | time4 | long double | time4>0 | Простая | Выход |
| MyForm.h | countCic2 | int | countCic2>=0 | Простая | Выход |
| MyForm.h | inputText | String^ | Строка без пробелов | Структура | Ввод  Выход |
| MyForm.h | ptr | IntPtr | Адрес в памяти | Структура | Преобразование String^ в char |
| MyForm.h | slovo | string | Строка без пробелов | Структура | Хранение текста в формате string |
| MyForm.h | nachPol | int | nach>=0 | Простая | Индекс символов |
| MyForm.h | konPol | int | kon>=0 | Простая | Индекс символов |
| MyForm.h | isPalindrome | bool | true или false | Простая | Хранение результата функции |

# 2.5 Декомпозиция функции

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Название | Имя | Параметры | | Процедура/ функция | Тестируется/не тестируется |
| Ввод: тип | Вывод: тип |
| Рекурсивная функция 1 | resRec1 | n1: int  count1: int  timeRec1: long double | res1: long long | Функция | Тестируется |
| Циклическая функция 1 | resCic1 | n2: int  count2: int  timeCic1: long double | res2: long long | Функция | Тестируется |
| Рекурсивная функция 2 | resRec2 | n3: int  count3: int  timeRec2: long double | res3: long long | Функция | Тестируется |
| Циклическая функция 2 | resCic2 | n4: int  count4: int  timeCic2: long double | res4: long long | Функция | Тестируется |
| Проверка на палиндром | resPol | slovo: string | bool | Функция | Тестируется |

# 2.6 Алгоритмизация

Алгоритмы функций:

Рекурсивная функция 1 математической формулы:

- Начало замера времени;

- Увеличение значения счётчика вызовов;

- Если значение n1 уже есть в кеше (memo), возвращается сохранённое значение;

- Если n1 <= 1, возвращает 1;

- Если n1 чётное, значение вычисляется рекурсивно для n1 - 1, затем результат умножается на 2 и сохраняется в res1;

- Если n1 нечётное, значения вычисляются рекурсивно для n1 - 1 и n1 - 2, затем идёт сложение 3 с умножением значений при n1 - 1 и n1 - 2 и вычитание данного выражения этими значениями поочерёдно, затем сохраняется в res1;

- Сохранение значения в кеше;

- Замер времени окончен;

- Возврат значения.

Циклическая функция 1 математической формулы:

- Начало замера времени;

- Увеличение значения счётчика вызовов;

- Если n2 <= 1, возвращает 1;

- Инициализация переменных x и y, которые равны соответственно 1 и 2;

- Для всех чисел от 2 до n2, если число чётное, то результат будет равен умножению 2 на переменную x, иначе если число не чётное, то результат равен сложению 3 с умножением переменных x,y и затем вычитанием этих переменных поочерёдно, значение сохраняется в res2;

- Обновление переменных;

- Замер времени окончен;

- Возврат значения.

Рекурсивная функция 2 математической формулы:

- Начало замера времени;

- Увеличение значения счётчика вызовов;

- Если значение n3 уже есть в кеше (memo), возвращается сохранённое значение;

- Если n3 <= 0, возвращает 0;

- Если n3 = 1, возвращается 1;

- Если n3 = 2, возвращается 2;

- Если n3 чётное, значение вычисляется рекурсивно для n3 - 3, затем идёт сложение умножения 7 с n3 и значения при n3 - 3, также данное выражение делится на 9, затем сохраняется в res3;

- Если n3 является не чётным, то значения вычисляются рекурсивно для n3 - 1, n3 - 2, затем идёт умножения 5 на n3 и сложение значений при n3 – 1,n3 – 2, далее данное выражение делится на 7 и сохраняется в res3;

- Сохранение значения в кеше;

- Замер времени окончен;

- Возврат значения.

Циклическая функция 2 математической формулы:

- Начало замера времени;

- Увеличение значения счётчика вызовов;

- Если n4 = 1, возвращается 1;

- Если n4 = 2, возвращается 2;

- Инициализация переменных x и y, которые равны соответственно 1 и 2;

- Для всех чисел от 3 до n4, если число чётное, то результат будет равен умножению 7 на i (число, которое равно изначально 3, постепенно увеличивающееся до числа, который вводит пользователь), сложению этого выражения с х и делением всего значения на 9, если число не чётное, то результат будет равен умножению 5 на i, сложению с x, y поочерёдно и деление всего выражения на 7,сохранение значения в res4;

- Обновление переменных;

- Замер времени окончен;

- Возврат значения.

Проверка на палиндромность:

- Если нулевой индекс, хранящийся в переменной nach строки, которую будет вводить пользователь, больше или равен индексу конца, который хранится в переменной kon, возвращается true;

- Если символы по индексам переменных nach и kon не равны, возвращается false;

- Рекурсивно вызывается функция с увеличением переменной nach и уменьшением kon.

Алгоритм программы

Главный интерфейс:

- Создание главного окна;

- Добавление элементов управления:

- Поле для ввода;

- Поле для вывода;

- Основные кнопки, которые отвечают за функционал программы;

- Добавление обработчиков событий к кнопкам.

Обработка нажатий на кнопки:

1) Проверка на палиндром.

- Считывание строки, которую вводит пользователь;

- Проверка не пустая ли строка;

- Преобразование строки в string;

- Сравнение строки с помощью функции;

- Вывод результата “YES” или ”NO”.

2) Вычисление 1 математической формулы рекурсией.

- Считывание числа от пользователя;

- Проверка диапазона в пределе от 1 до 2500;

- Запуск рекурсивной функции resRec1;

- Если число функции выходит за диапазон выводится ошибка;

- Показать результат.

3) Вычисление 1 математической формулы циклом.

- Считывание числа от пользователя;

- Проверка диапазона в пределе от 1 до 2.100.000.000;

- Запуск циклической функции resCic1;

- Если число функции выходит за диапазон выводится ошибка;

- Показать результат.

4) Вычисление 2 математической формулы рекурсией.

- Считывание числа от пользователя;

- Проверка диапазона в пределе от 1 до 5000;

- Запуск рекурсивной функции resRec2;

- Если число функции выходит за диапазон выводится ошибка;

- Показать результат.

5) ) Вычисление 1 математической формулы циклом.

- Считывание числа от пользователя;

- Проверка диапазона в пределе от 1 до 2.100.000.000;

- Запуск циклической функции resCic2;

- Если число функции выходит за диапазон выводится ошибка;

- Показать результат.

# Приложение А

Изображение выглядит как текст, снимок экрана, Шрифт, графический дизайн

Контент, сгенерированный ИИ, может содержать ошибки.

Рисунок 1 – Блок-схема рекурсивной функции 1 математической формулы

Изображение выглядит как текст, снимок экрана, дизайн

Контент, сгенерированный ИИ, может содержать ошибки.

Рисунок 2 – Блок-схема рекурсивной функции 2 математической формулы

Изображение выглядит как текст, снимок экрана, Шрифт, графический дизайн

Контент, сгенерированный ИИ, может содержать ошибки.

Рисунок 3 – Блок-схема циклической функции 1 математической формулы

Изображение выглядит как текст, снимок экрана, черно-белый, дизайн

Контент, сгенерированный ИИ, может содержать ошибки.

Рисунок 4 – Блок-схема циклической функции 2 математической формулы