###### МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ РОССИЙСКОЙ ФЕДЕРАЦИИ

###### ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ АВТОНОМНОЕ ОБРАЗОВАТЕЛЬНОЕ УЧРЕЖДЕНИЕ ВЫСШЕГО ОБРАЗОВАНИЯ

###### НОВОСИБИРСКИЙ НАЦИОНАЛЬНЫЙ ИССЛЕДОВАТЕЛЬСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ

###### Факультет информационных технологий

**Кафедра параллельных вычислений**

ОТЧЕТ

О ВЫПОЛНЕНИИ ЛАБОРАТОРНОЙ РАБОТЫ

«ИЗУЧЕНИЕ ОПТИМИЗИРУЮЩЕГО КОМПИЛЯТОРА»

студента 2 курса, 22202 группы

**Бальчинова А.С.**

Направление 09.03.01 – «Информатика и вычислительная техника»

Преподаватель:

В.А. Перепёлкин

Новосибирск 2023

**СОДЕРЖАНИЕ**

[ЦЕЛЬ 3](#_Toc18443921)

[ЗАДАНИЕ 3](#_Toc18443922)

[ОПИСАНИЕ РАБОТЫ 4](#_Toc18443923)

[ЗАКЛЮЧЕНИЕ 5](#_Toc18443924)

[Приложение. 6](#_Toc18443925)

# ЦЕЛИ

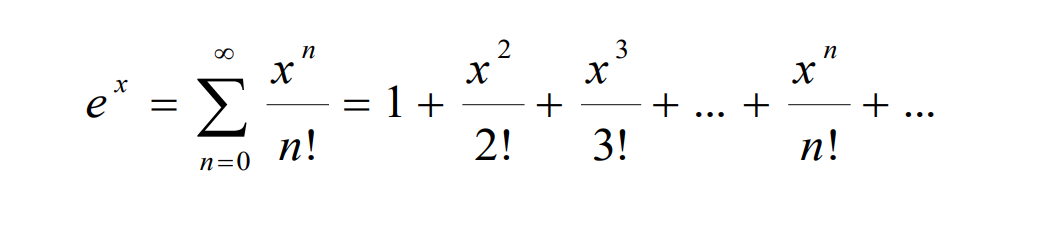
1.Изучение основных функций оптимизирующего компилятора, и некоторых примеров оптимизирующих преобразований и уровней оптимизации.

2. Получение базовых навыков работы с компилятором GCC.

3. Исследование влияния оптимизационных настроек компилятора GCC на время исполнения программы.

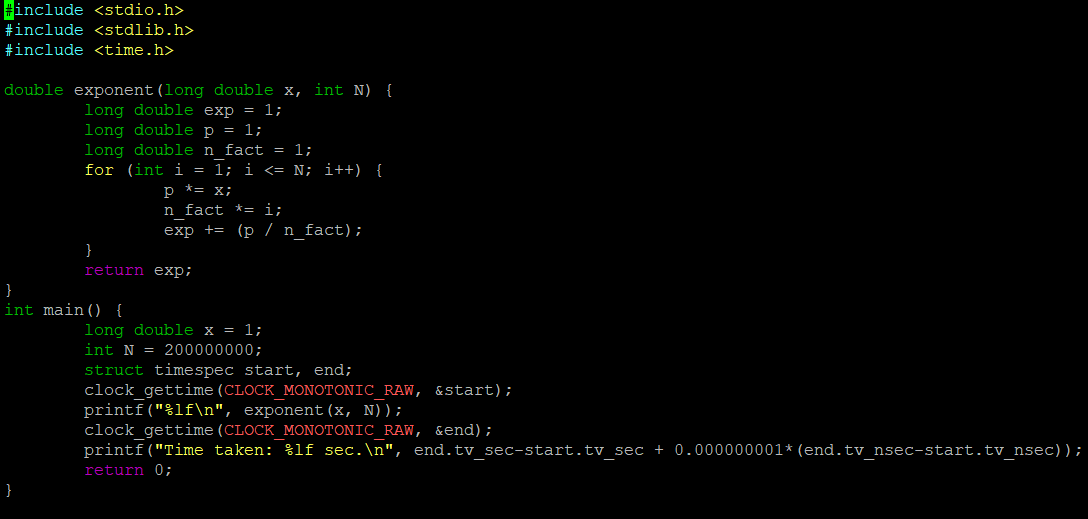
# ЗАДАНИЕ

Алгоритм вычисления функции ex с помощью разложения в ряд Маклорена по первым N членам этого ряда:

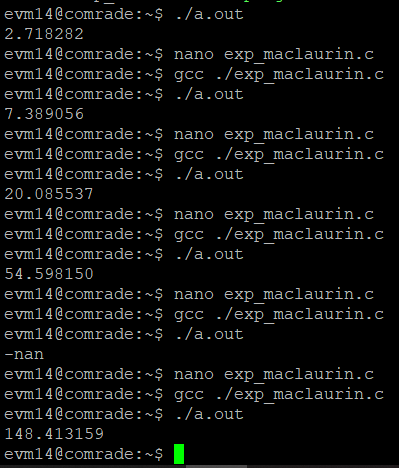


Область сходимости ряда: − ∞ ≤ x ≤ ∞ .

# ОПИСАНИЕ РАБОТЫ

1. Реализация алгоритма из задания на языке программирования C. 
2. Проверка правильности работы программы на нескольких тестовых наборах входных данных.
3. x = 1, N = 80000, ответ e^x = e^1 = 2.718282
4. x = 2, N = 80000, ответ e^x = e^2 = 7.389056
5. x = 3, N = 80000, ответ e^x = e^3 = 20.085537
6. x = 4, N = 80000, ответ e^x = e^4 = 54.598150

Программа выдаёт правильный ответ на нескольких тестовых значениях x, от N зависит точность вычислений.



1. Выбор значение параметра N таким, чтобы время работы программы было порядка 30-60 секунд.

Экспериментально (подбором) были выбраны значения N1 = 335,000,000, при котором время работы программы (при запуске “без оптимизаций”) составляет примерно 60,35 секунд, и N2 = 170,000,000, при котором время работы программы составляет примерно 30,75 секунд.

1. Программу скомпилировать компилятором GCC с уровнями оптимизации -O0, -O1, -O2, -O3, -Os, -Ofast, -Og под архитектуру процессора x86.

Пример используемого ключа компилятора:

gcc –O0 lab2.c -o lab2.bin –Wall

1. Для каждого из семи вариантов компиляции измерить время работы программы при нескольких значениях N.

|  |  |  |
| --- | --- | --- |
|  | **N1 = 335\*10^6** | **N2 = 170\*10^6** |
| **O0** | 60,348332 | 30,750945 |
| **O1** | 56,670892 | 28,895220 |
| **O2** | 55,833616 | 28,455900 |
| **O3** | 55,835369 | 28,458459 |
| **Os** | 55,832880 | 28,462556 |
| **Ofast** | 55,837167 | 28,457983 |
| **Og** | 56,359554 | 28,731146 |

Можно заметить, что в данном случае оптимизации O0 заметно медленнее O1, O2 и др. Разница во времени выполнения между остальными ключами оптимизации не превышает 1 секунды.

**ЗАКЛЮЧЕНИЕ**

Были изучены основные функции оптимизирующего компилятора, и некоторые примеры оптимизирующих преобразований и уровней оптимизации.

Получены базовые навыки работы с компилятором GCC.

Было исследовано влияние оптимизационных настроек компилятора GCC на время исполнения программы.

# Приложение. *Листинг программы*

#include <stdio.h>

#include <stdlib.h>

#include <time.h>

double exponent(double x, int N) {

double exp = 1;

double p = 1;

double n\_fact = 1;

for (int i = 1; i <= N; i++) {

p \*= x;

n\_fact \*= i;

exp += (p / n\_fact);

}

return exp;

}

int main() {

long double x = 1;

int N = 170000000;

struct timespec start, end;

clock\_gettime(CLOCK\_MONOTONIC\_RAW, &start);

printf("%lf\n", exponent(x, N));

clock\_gettime(CLOCK\_MONOTONIC\_RAW, &end);

printf("Time taken: %lf sec.\n", end.tv\_sec-start.tv\_sec + 0.000000001\*(end.tv\_nsec-start.tv\_nsec));

return 0;

}