Министерство общего и профессионального

образования Российской Федерации

Пермский государственный технический университет

Лабораторная работа №7

"Рекурсия"

Вариант №12

Выполнила студентка группы РИС 23-3б:

Федорова О.И.

Пермь 2024

1. Постановка задачи. Посчитать сумму ряда с n-нным количеством элементов методом рекурсии и итераций, сравнить: где получилось точнее и быстрее.

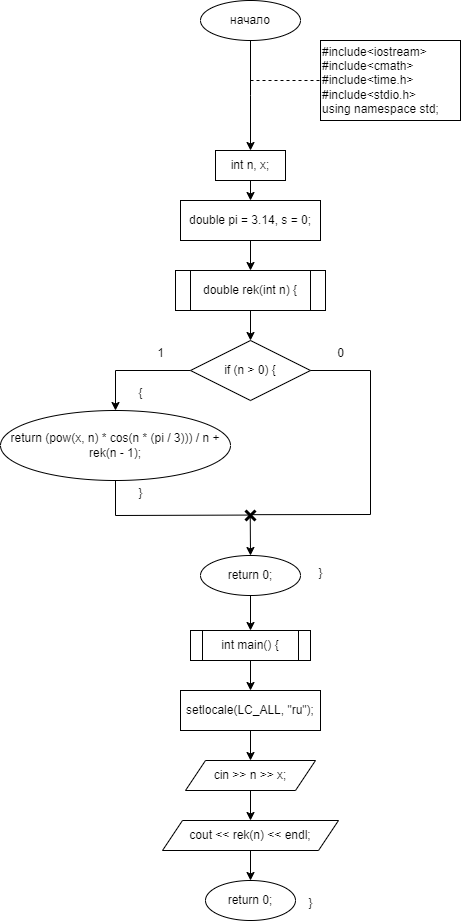
**Метод рекурсии**

2.1 Анализ задачи.

1) Следует начать подсчет суммы начиная с последнего n-нного элемента, уменьшая каждый раз n.

2) Создание рекурсивной функции, которая будет каждый раз вызывать сама себя, но на единицу меньше пока n > 0.

3) После вызова рекурсивной функции в функции main на экран выведется вся сумма ряда элементов.

3.1 Блок-схема.

4.1 Код программы.

#include<iostream> // подключение стандартной библиотеки

#include<cmath> // подключение библиотеки для возведения в степень

#include<time.h> // подключение библиотеки для измерения времени работы программы

#include<stdio.h> // подключение библиотеки для измерения времени работы программы

using namespace std; // подключение пространства имен

int n, x; // резервирование памяти для n- окончательное число последовательности, x- коэффициент

double pi = 3.14, s = 0; // объявление переменной pi- число пи, s-сумма последовательности, изначально равна 0

double rek(int n) { // создание рекурсивной функции для подсчета суммы последовательности

if (n > 0) { // условие для окончания суммирования элементов

return (pow(x, n) \* cos(n \* (pi / 3))) / n + rek(n - 1); // возврат значения суммы с обращением к функции с другим элементу

}

return 0; // завершение работы функции

}

int main() { // создание функции, которая будет выполнять задачу

/\*clock\_t start = clock();\*/ // начало отсчета времени

setlocale(LC\_ALL, "ru"); // функция для подключения русского языка

cout << "Введите количество элементов и коэффициент: "; // вывод пояснения для пользователя

cin >> n >> x; // ввод n и x

cout << "Сумма ряда: "; // вывод пояснения для пользователя

cout << rek(n) << endl; // вывод результата функции rek

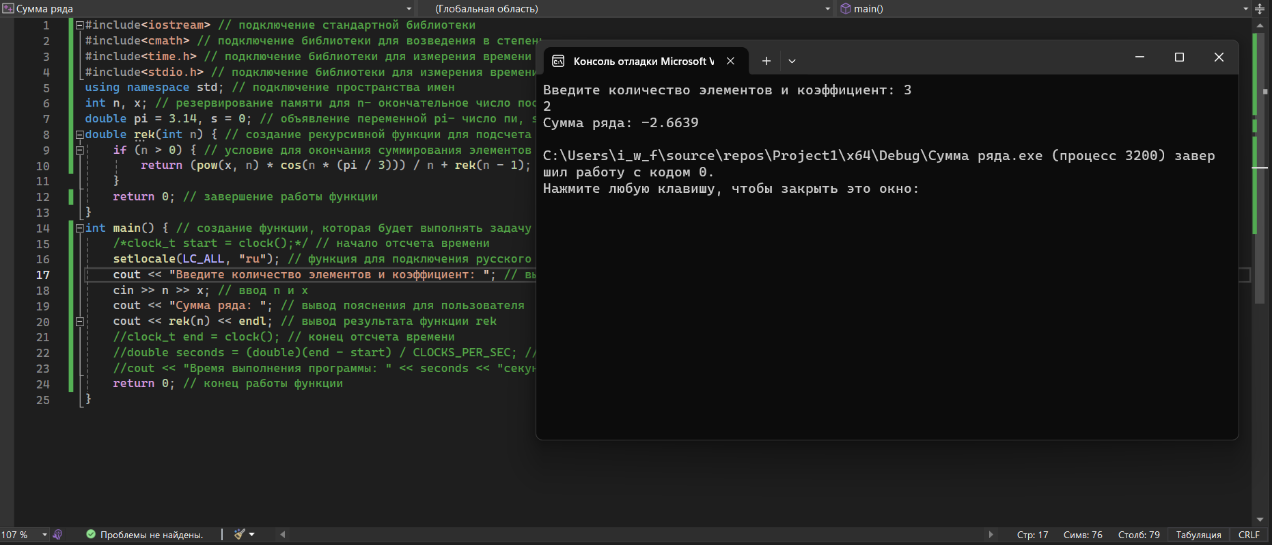
//clock\_t end = clock(); // конец отсчета времени

//double seconds = (double)(end - start) / CLOCKS\_PER\_SEC; // вычисление времени выполнения программы

//cout << "Время выполнения программы: " << seconds << "секунд"; // вывод времени выполнения программы

return 0; // конец работы функции

}

5.1 Результат работы программы.

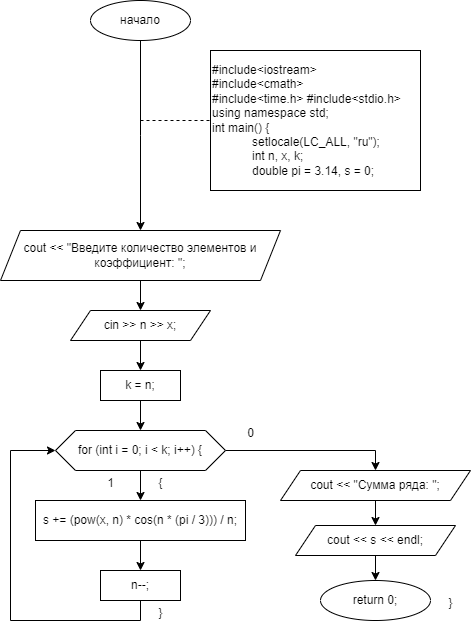
**Метод итераций.**

2.2 Анализ задачи.

1) Следует начать подсчет суммы начиная с последнего n-нного элемента, уменьшая каждый раз n.

2) Создание цикла for, который с помощью счетчика i будет повторяться ровно n раз, и каждый раз к сумме(изначально равна 0) прибавлять элемент ряда от наибольших значений n к наименьшим(n уменьшается).

3) После выполнения цикла на экран выведется вся сумма ряда элементов.

3.2 Блок-схема.

4.2 Код программы.

#include<iostream> // подключение стандартной библиотеки

#include<cmath> // подключение библиотеки для возведения в степень

#include<time.h> // подключение библиотеки для измерения времени работы программы

#include<stdio.h> // подключение библиотеки для измерения времени работы программы

using namespace std; // подключение пространства имен

int main() { // создание функции, которая будет выполнять задачу

/\*clock\_t start = clock();\*/ // начало отсчета времени

setlocale(LC\_ALL, "ru"); // функция для подключения русского языка

int n, x, k; // резервирование памяти для n- окончательное число последовательности, x- коэффициент, k- сохранит значение n

double pi = 3.14, s = 0; // объявление переменной pi- число пи, s-сумма последовательности, изначально равна 0

cout << "Введите количество элементов и коэффициент: "; // вывод пояснения для пользователя

cin >> n >> x; // ввод n и x

k = n; // k понадобится чтобы сохранить значение n, так как n будет меняться

for (int i = 0; i < k; i++) { // цикл для прохождения по всем элементам последовательности

s += (pow(x, n) \* cos(n \* (pi / 3))) / n; // с каждым шагом подсчитывает сумму с новым значением n

n--; // изменение n для подсчета суммы

}

cout << "Сумма ряда: "; // вывод пояснения для пользователя

cout << s << endl; // вывод полученной суммы

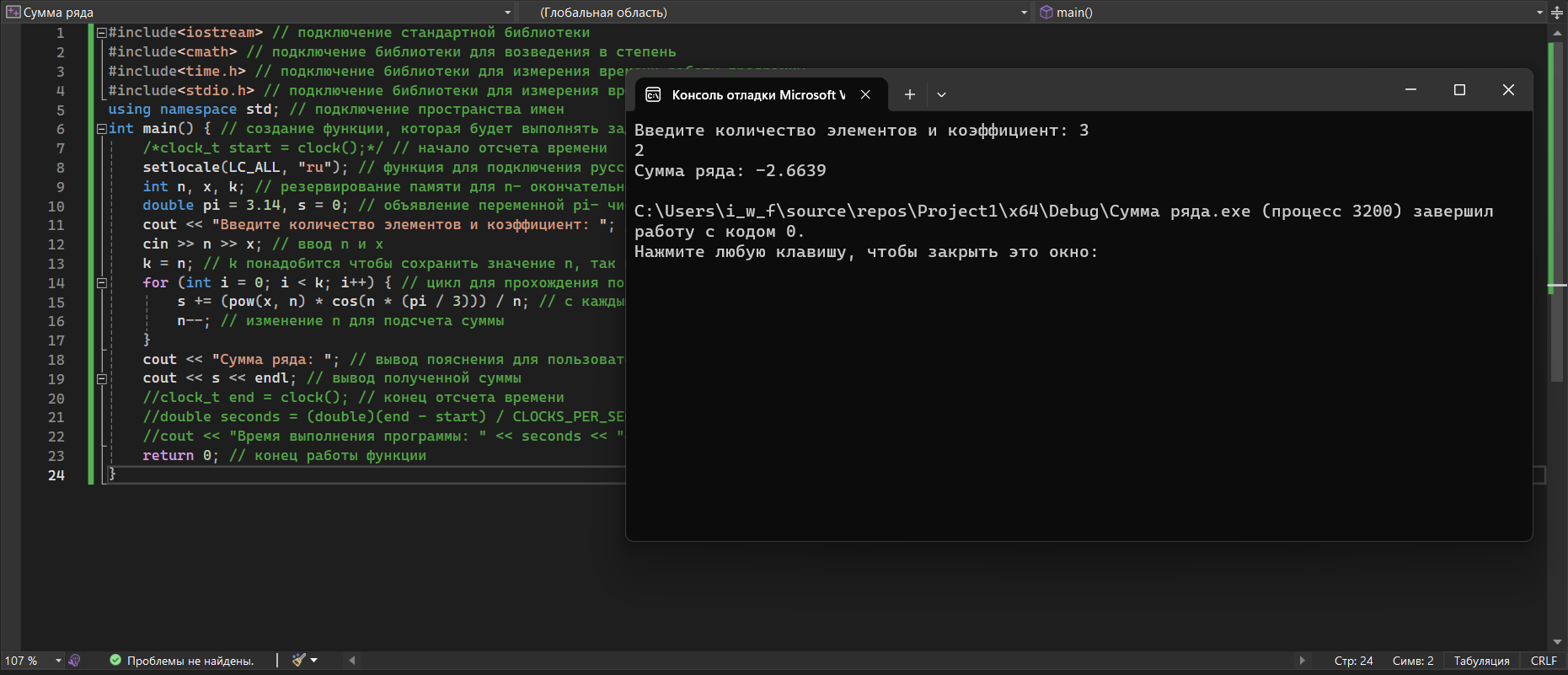
//clock\_t end = clock(); // конец отсчета времени

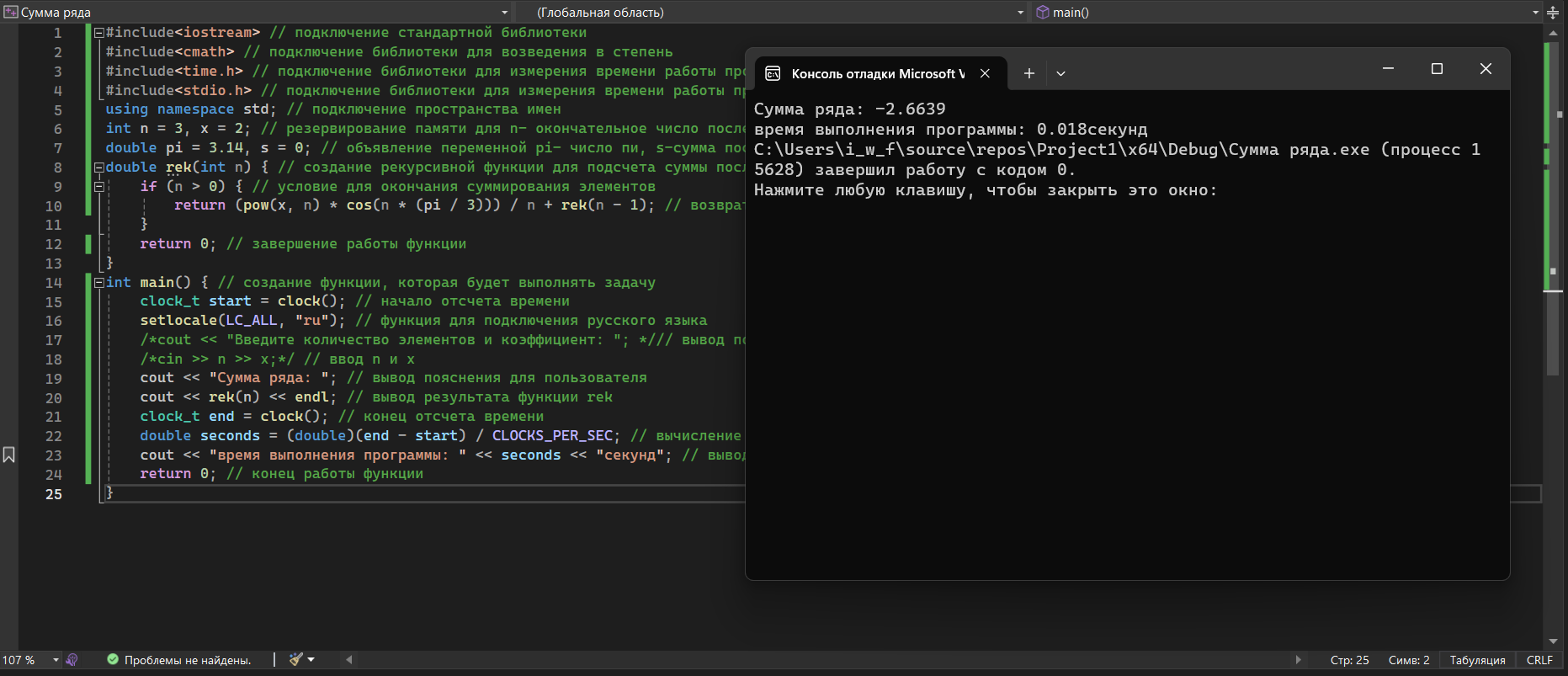
//double seconds = (double)(end - start) / CLOCKS\_PER\_SEC; // вычисление времени выполнения программы

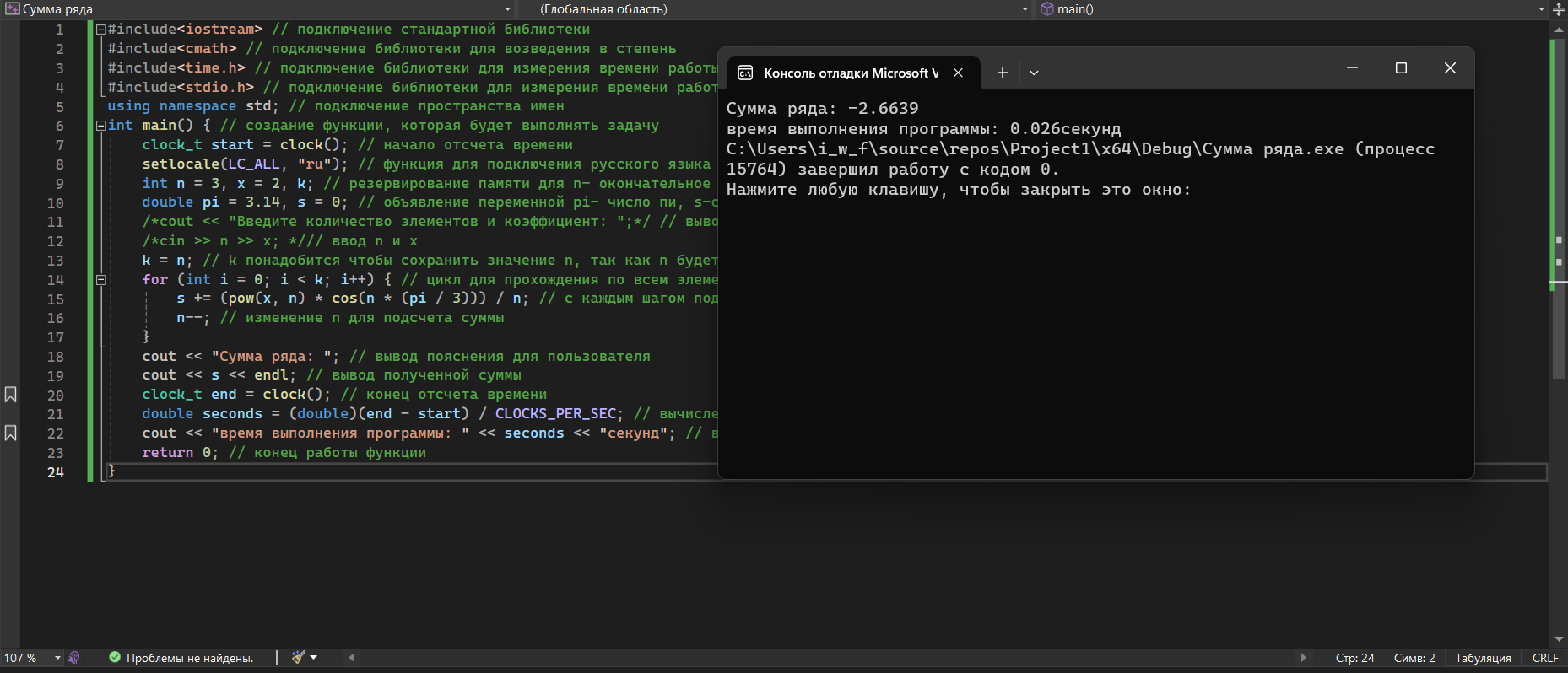
//cout << "Время выполнения программы: " << seconds << "секунд"; // вывод времени выполнения программы

return 0; // конец работы функции

}

5.2 Результат работы программы.

6. Выводы: точность ответов одинакова в обоих методах, а скорость выше у метода рекурсии на 8 миллисекунд.

Время выполнения методом рекурсии: 0,018 секунд или 18 миллисекунд.

Время выполнения методом итераций: 0,026 секунд или 26 миллисекунд.

Ссылка на работу в GitHub: <https://github.com/slurree/first.git>