Харьковский национальный университет имени В. Н. Каразина

Факультет компьютерных наук

Отчет по лабораторной работе №4

По дисциплине: «Алгоритмизация и программирование»

**«Циклы с вложением»**

Выполнил: студент группы КС-11

Безрук Ю. Р.

Проверил: преп. каф. ПОИИ

  Севидов Сергей Михайлович

Харьков

2018

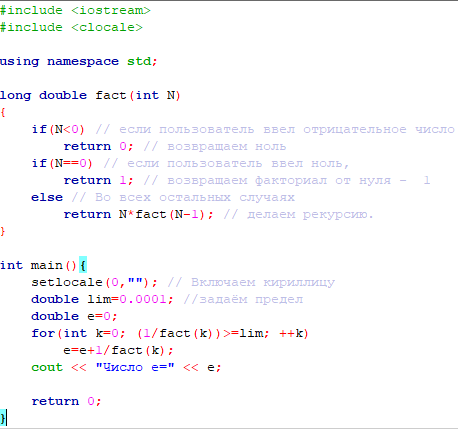
**ЗАДАНИЕ 1**

Напишите программу для вычисления числа e по формуле:

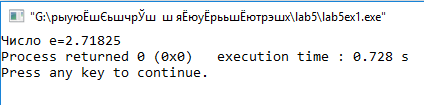
****

Вычисление суммы продолжать до тех пор, пока очередное слагаемое по модулю не станет меньше 10^-4 .

**ИСХОДНЫЙ КОД**

****

**ПРИМЕР ВЫПОЛНЕНИЯ**

****

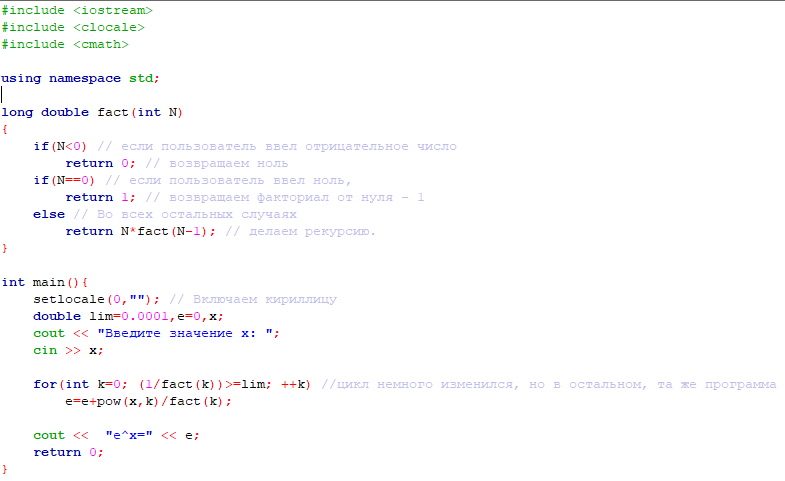
**ЗАДАНИЕ 2**

Напишите программу для вычисления значения экспоненциальной функции ex в заданной точке x по формуле:

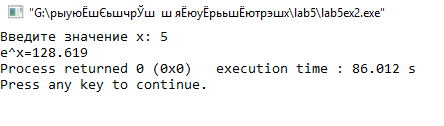


Вычисление суммы продолжать до тех пор, пока очередное слагаемое по модулю не станет меньше 10^-4. Конкретное значение x вводится с клавиатуры.

**ИСХОДНЫЙ КОД**



**ПРИМЕР ВЫПОЛНЕНИЯ**

****

**ЗАДАНИЕ 3**

Табулировать экспоненциальную функцию e^x для значений [0,2] x с шагом 0.1. Результат вывести в виде следующей таблицы:

x sum exp delta

0 1 1 0

0.1 1.1052 1.105171 0.000029

0.2 \_\_\_\_\_\_ \_\_\_\_\_\_ \_\_\_\_\_\_

…

\_\_\_\_\_\_ \_\_\_\_\_\_ \_\_\_\_\_\_

2 \_\_\_\_\_\_ \_\_\_\_\_\_ \_\_\_\_\_\_

**ИСХОДНЫЙ КОД**

#include <iostream>

#include <clocale>

#include <cmath>

using namespace std;

long double fact(int N){

if(N<0) // если пользователь ввел отрицательное число

return 0; // возвращаем ноль

if(N==0) // если пользователь ввел ноль,

return 1; // возвращаем факториал от нуля - 1

else // Во всех остальных случаях

return N\*fact(N-1); // делаем рекурсию.

}

long double sum(double x, double lim){ //функция sum возвращает значение экспоненты в заданной точке

double e=0;

for(int k=0; (pow(x,k)/fact(k))>=lim; ++k)

e=e+pow(x,k)/fact(k);

return e;

}

long double delta(float x,double lim){//фукция delta возвращает значение разности моей и встроенной функций экспоненты

return exp(x)-sum(x,lim);

}

int main(){

setlocale(0,""); // Включаем кириллицу

double lim;

cout << "Введите значение погрешности: ";

cin >> lim;

cout << "x\tsum\texp\tdelta\n-------------------------------\n"; //верхушка таблицы

for (float x=0; x<=2; x+=0.1){ //в цикле просто поочередно вызываем тр функции для значей таблицы

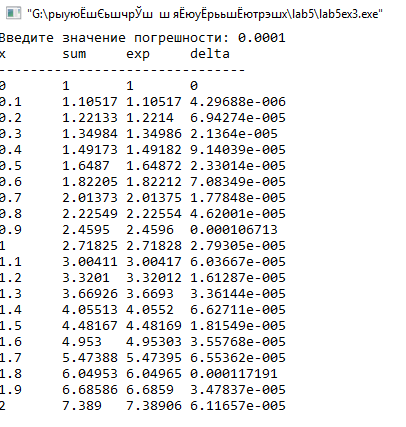
cout << x <<"\t"<< sum(x,lim) <<"\t"<< exp(x) <<"\t"<< delta(x,lim) << endl;

}

return 0;

}

**ПРИМЕР ВЫПОЛНЕНИЯ**

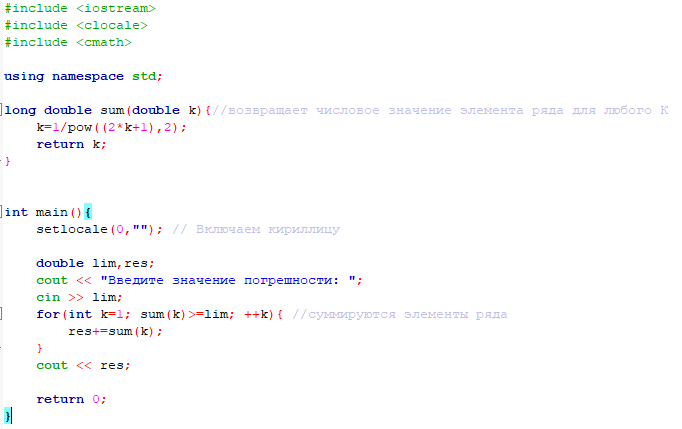
****

**ЗАДАНИЕ 4**

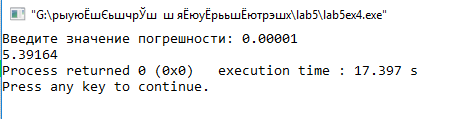
6)Вычислить бесконечную сумму с заданной (с клавиатуры) точностью E (E>0). Считать, что требуемая точность достигнута, если очередное слагаемое оказалось по модулю меньше, чем E:



**ИСХОДНЫЙ КОД**

****

**ПРИМЕР ВЫПОЛНЕНИЯ**

****

**ЗАДАНИЕ 5**

Воспользуйтесь программой из «Задания 3» лабораторной работы №4 (поиск максимальной степени числа, не превосходящей заданного N), а также блок-схемой из задания 4 лабораторной работы №2 (запись заданного числа в произвольной системе счисления) для того, чтобы написать полноценную программу перевода числа в систему счисления по основанию k. На входе программы — число для перевода N и основание системы счисления k. На выходе представление введенного числа N в k-чной системе счисления. Программа должна работать при значениях k от 2 до 16 включительно.

**ИСХОДНЫЙ КОД**

**ПРИМЕР ВЫПОЛНЕНИЯ**

#include <iostream>

#include <clocale>

#include <cmath>

using namespace std;

int Mpow(int k,int m){//возвращает максимальную степень числа k, в которой оно не превышает m

int res=0;

while(pow(k,res)<=m)

++res;

return res-1;

}

int main(){

setlocale(0,""); // Включаем кириллицу

int N,k,n,d;

cout << "Введите число: ";

cin >> N;

cout << "Введите основание системы счисления для перевода(от 2 до 16): ";

cin >> k;

if((k<2)||(k>16)){

cout << "Некорректное значение основания системы исчисления!!!!";

return 0;

}

n=Mpow(k,N);

while(n>=0){

d=N/pow(k,n);

if(d>=10){

switch(d){

case 10: cout << "a"; break;

case 11: cout << "b"; break;

case 12: cout << "c"; break;

case 13: cout << "d"; break;

case 14: cout << "e"; break;

case 15: cout << "f"; break;

}

}else

cout << d;

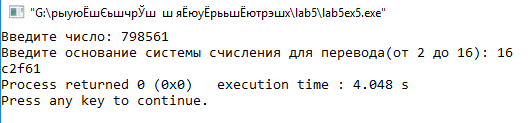
N=N-d\*pow(k,n);

n=n-1;

}

return 0;

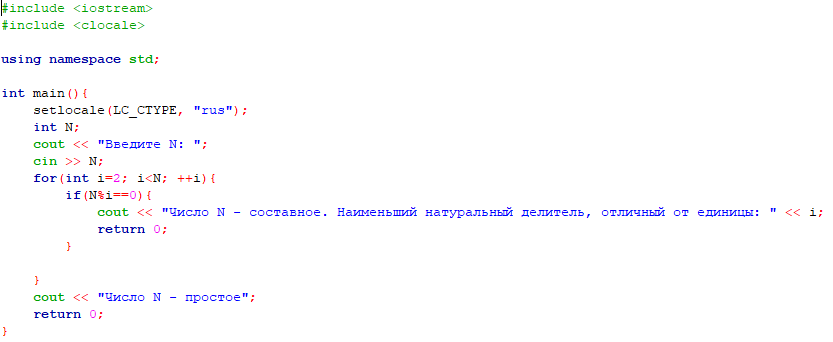
}

****

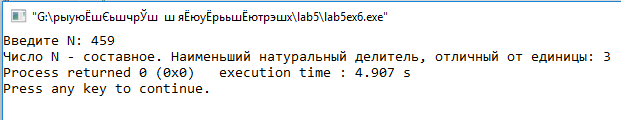
**ЗАДАНИЕ 6**

Написать программу для проверки натурального числа N на простоту. N вводится с клавиатуры.

**ИСХОДНЫЙ КОД**



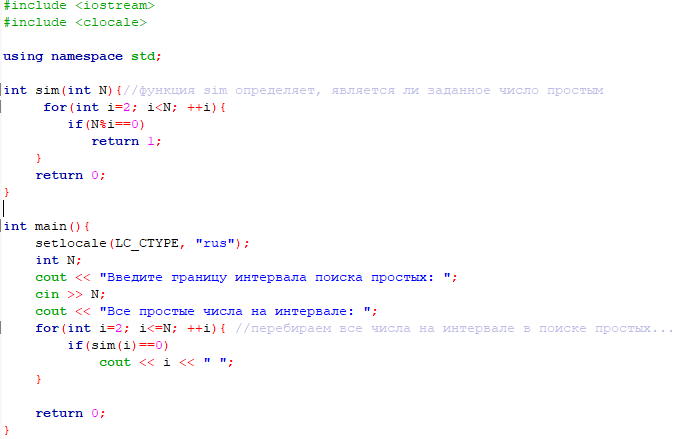
**ПРИМЕР ВЫПОЛНЕНИЯ**

****

**ЗАДАНИЕ 7**

Написать программу для поиска всех простых чисел на интервале 2..N. N вводится с клавиатуры.

**ИСХОДНЫЙ КОД**

****

**ПРИМЕР ВЫПОЛНЕНИЯ**

