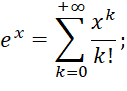
|  |  |
| --- | --- |
| Министерство науки и высшего образования Российской Федерации  Санкт-Петербургский политехнический университет Петра Великого  Институт компьютерных наук и технологий  Высшая школа киберфизических систем и управления | |
| ОТЧЁТ  по дисциплине «Теория и технология программирования»  по лабораторной работе №4(9) | |
| **Выполнил**: |  |
| студент гр. 3530902/10003 | Сурьянинова В. В. |
|  |  |
|  |  |
| **Проверил:**  ассистент |  |
|  | Журавская А. |
|  |  |
|  | |

1. Задание

Написать программу, в которой для вычисления функции, используется динамически подключенная библиотека dll.

Вариант №28:



1. Текст программы

Написать программу

Код библиотеки dll, "Source.cpp":

#include <iostream>

#include "math.h"

**using** **namespace** std;

**long** **double** factorial(**int** n) {

**if** ((n == 0) || (n == 1)) {

**return** 1;

}

**else** {

**return** n \* factorial(n - 1);

}

}

/\*long double power(long double x, int k) {

if (k == 0) {

return 1;

}

else {

return x \* power(x, k - 1);

}

}\*/

**extern** "C" \_\_declspec(dllexport)

**long** **double** f(**long** **double** x, **long** **double** e) {

**long** **double** sum = 0, max, next = 0;

**for** (**int** k = 0; abs((pow(x, k + 1) / factorial(k + 1)) - (pow(x, k)) / factorial(k)) > e; k++)

{

sum += (pow(x, k)) / factorial(k);

}

**return** sum;

}

**extern** "C" \_\_declspec(dllexport)

**long** **double** F(**long** **double** x) {

**long** **double** res;

res = exp(x);

**return** res;

}

Код библиотеки dll, "Header.h":

#pragma once

**extern** "C" \_\_declspec(dllexport) **long** **double** f(**long** **double**, **long** **double**);

**extern** "C" \_\_declspec(dllexport) **long** **double** F(**long** **double**);

Основной код. "main.cpp":

#include <iostream>

#include "math.h"

#include <Windows.h>

#include <stdlib.h>

**using** **namespace** std;

**long** **double** residual(**long** **double** a, **long** **double** b){

**return** sqrt(abs(pow(a, 2) - pow(b, 2)));

}

**long** **double** enter(string& k, string c) {

**bool** f = **true**;

**int** count = 0, dot = 0, min = 0;

**while** (f == **true**) {

cout << "enter the " << c << ": ";

k = "";

cin >> k;

**int** l = k.length();

count = 0, dot = 0, min = 0;

**for** (**int** i = 0; i < k.length(); i++) {

**if** ((k[i] >= '0') && (k[i] <= '9')) {

count++;

}

**if** (k[i] == '.') {

dot += 1;

}

**if** (k[0] == '-') {

min = -1;

}

**else** {

min = 1;

}

}

**if** (dot == 1) {

l -= 1;

}

**if** (min == -1) {

l -= 1;

}

**if** (count == l) {

f = **false**;

}

}

**int** t1 = 0, fl = 0, t2 = 0, u = 0;

**long** **double** res = 0;

**if** (min == -1) {

**for** (**int** i = 1; i < k.length(); i++) {

**if** ((fl == 0) && (k[i] == '.')) {

fl = 1;

}

**else** **if** (fl == 1) {

t2 = 10 \* t2 + k[i] - 48;

u -= 1;

}

**else** **if** (fl == 0) {

t1 = 10 \* t1 + k[i] - 48;

}

}

}

**else** {

**for** (**int** i = 0; i < k.length(); i++) {

**if** ((fl == 0) && (k[i] == '.')) {

fl = 1;

}

**else** **if** (fl == 1) {

t2 = 10 \* t2 + k[i] - 48;

u -= 1;

}

**else** **if** (fl == 0) {

t1 = 10 \* t1 + k[i] - 48;

}

}

}

res = (t1 + t2 \* pow(10, u)) \* min;

**return** res;

}

**int** main() {

HINSTANCE mydll = LoadLibraryA("DinamicLibrary.dll");

**if** (mydll == **NULL**) {

cout << "ERROR";

**return** 0;

}

**typedef** **long** **double** (\*Afun)(**long** **double**, **long** **double**);

Afun f = (Afun)GetProcAddress(mydll, "f");

**typedef** **long** **double** (\*Bfun)(**long** **double**);

Bfun F = (Bfun)GetProcAddress(mydll, "F");

**long** **double** e, xstart, xend, dx, xideal, i;

string k;

e = enter(k, "e");

xstart = enter(k, "xstart");

xend = enter(k, "xend");

dx = enter(k, "dx");

**if** ((xstart > xend) && (dx > 0)) {

cout << "wrong input, dx must be < 0" << endl;

exit(0);

}

**if** ((xstart < xend) && (dx < 0)) {

cout << "wrong input, dx must be > 0" << endl;

exit(0);

}

printf("|---------|---------------------------|----------------------|----------------------|\n");

printf("| x | f(x) | F(x) | residual |\n");

printf("|---------|---------------------------|----------------------|----------------------|\n");

**if** (xstart > xend) {

**for** (i = xstart; i >= xend; i += dx) {

printf("| %7.4f | %25.5lf | %20.5lf | %20.5lf |\n", i, f(i, e), F(i), residual(f(i, e), F(i)));

printf("|---------|---------------------------|----------------------|----------------------|\n");

}

}

**else** {

**for** (i = xstart; i <= xend; i += dx) {

printf("| %7.4f | %25.5lf | %20.5lf | %20.5lf |\n", i, f(i, e), F(i), residual(f(i, e), F(i)));

printf("|---------|---------------------------|----------------------|----------------------|\n");

}

}

**if** ((xstart > xend) && (i < xend) && (i + dx != xend)) {

printf("| %7.4f | %25.5lf | %20.5lf | %20.5lf |\n", xend, f(xend, e), F(xend), residual(f(xend, e), F(xend)));

printf("|---------|---------------------------|----------------------|----------------------|\n");

}

**if** ((xstart < xend) && (i > xend) && (i - dx != xend)) {

printf("| %7.4f | %25.5lf | %20.5lf | %20.5lf |\n", xend, f(xend, e), F(xend), residual(f(xend, e), F(xend)));

printf("|---------|---------------------------|----------------------|----------------------|\n");

}

xideal = enter(k, "xideal");

printf("|-----------|---------------------------|----------------------|----------------------|\n");

printf("| e | f(xideal) | F(xideal) | residual |\n");

printf("|-----------|---------------------------|----------------------|----------------------|\n");

**for** (**long** **double** i = 0.1; i >= 0.0000001; i \*= 0.1) {

printf("| %7.7f | %25.5lf | %20.5lf | %20.5lf |\n", i, f(xideal, i), F(xideal), residual(f(xideal, i), F(xideal)));

printf("|-----------|---------------------------|----------------------|----------------------|\n");

}

**return** 0;

}

1. Входные и выходные данные

Входные данные: точность e, начальный x (xstart) и конечный x (xend), шаг dx, идеальное x (xideal).

Выходные данные - две таблицы, в первой сравнивается значение рядной и стандартной функциях при x ∈ [xstart; xend] с шагом dx, во второй выводятся значения двух функций от xideal при измененной точности в пределах от 0.1 до 10-7.

1. Пример работы программы

На рисунке 3 представлен пример выполненной программы, где пользователь осуществил ввод значений с ошибками (буквы вместо цифр). На рисунке 4 представлен пример, где пользователь ввел некорректные данные, которые противоречат условию. На рисунке 5 пример корректного выполнения программы.

Изображение выглядит как текст

Автоматически созданное описание

Рисунок 3

Изображение выглядит как текст

Автоматически созданное описание

Рисунок 4

Изображение выглядит как текст

Автоматически созданное описание

Рисунок 5

1. Вывод

В результате проделанной мною работы я написала динамическую библиотеку dll, в которой указаны две функции, вычисляющие простую и рядную функции. Это очень полезный навык, который пригодится мне в будущем для программирования.