Лабораторная работа №4

Механизмы передачи параметров.

ЗАДАНИЕ

Вариант 6

1. Составить алгоритм и программу, реализующую разные механизмы передачи параметров, на примере нахождения значений функций.
   1. Функция принимает параметры по значению и возвращает значение;
   2. Функция в качестве параметров принимает указатели, в том числе и параметр адреса результата, а сама функция типа Void.
   3. Функция принимает параметры по ссылке и возвращает ссылку на результат.

Исходные данные представлены в таблице (по вариантам). Выбор механизма передачи параметров оформить с помощью оператора Switch.

Функция вычисляет сумму членов ряда. **Вычисление суммы необходимо выполнять до тех пор, пока очередное слагаемое (член ряда) по модулю не станет меньше заданной точности .**

|  |  |  |  |
| --- | --- | --- | --- |
| 6 |  | 1,7 |  |

ХОД РАБОТЫ

ЗАДАНИЕ 1

Результат работы программы представлен в соответствии с рисунком 1.

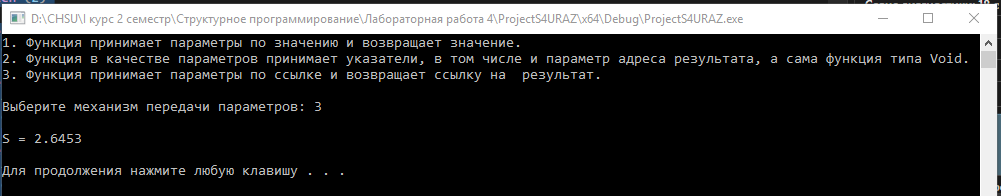


Рисунок 1 – результаты работы программы.

Текст программы:

#include <iostream>

#include <cmath>

using namespace std;

int fact(int n)

{

int k = 1;

for (int i = 1; i<=n; i++)

k \*= i;

return k;

}

float f(float x, int n)

{

return pow(x, 2 \* n + 1) / fact((2 \* n + 1));

}

float f1(float x, float e)

{

float s = 0;

float r;

**for (int n = 0; (abs(r = f(x,n))) >= e; n++)**

s += r;

return s;

}

void f2(float\* x, float\* s, float\* e)

{

float r;

for (int n = 0; (abs(r = f(\*x, n))) > \*e; n++)

\*s += r;

}

float f3(float& x, float& s, float &e)

{

float r;

for (int n = 0; (abs(r = f(x, n))) > e; n++)

s += r;

return s;

}

void main() {

setlocale(LC\_CTYPE, "rus");

cout << "1. Функция принимает параметры по значению и возвращает значение." << endl;

cout << "2. Функция в качестве параметров принимает указатели, в том числе и параметр адреса результата, а сама функция типа Void." << endl;

cout << "3. Функция принимает параметры по ссылке и возвращает ссылку на результат." << endl;

cout << endl << "Выберите механизм передачи параметров: ";

float e = pow(10, -3);

int z;

cin >> z;

float x = 1.7;

float res = 0;

switch (z)

{

case 1:

cout << endl << "S = " << f1(x,e) << endl;

break;

case 2:

f2(&x, &res, &e);

cout << endl << "S = " << res << endl;

break;

case 3:

cout << endl << "S = " << f3(x, res, e) << endl;

break;

default:

break;

}

cout << endl;

system("pause");

}

Алгоритм программы:

Пользователь выбирает механизм передачи параметров из предложенных вариантов переключателем switch-case.

Объявление функции fact для нахождения факториала числа n, полученного в качестве параметра данной функции. Для нахождения факториала в функции объявлена переменная k, значением которой является произведение всех чисел от 1 до n.

Объявление функции f типа float для вычисления значения члена ряда для полученных значений x и n в качестве параметров данной функции. Функция возвращает вычисленное значение.

Объявление функции f1 типа float для вычисления значения суммы членов ряда до тех пор, пока член ряда по модулю не станет меньше заданной точности **** в переменной e. Переменная s типа float необходима для хранения значения суммы членов ряда. До тех пор, пока член ряда (для удобства объявляем и сохраняем его значение в переменной r типа float) по модулю не меньше точности e, суммируем члены ряда в переменную s.

Аналогично функции f1 объявляется функция f2 типа void, которая в качестве параметров принимает указатели на адреса переменных n и res, значение переменной e.

Аналогично функции f1 объявляется функция f3 типа float, которая в качестве параметров принимает ссылки, производит вычисления и возвращает ссылку на результат.

Вывод:

В ходе выполнения лабораторной работы мы научились использовать механизмы передачи параметров и закрепили полученные навыки.

Указатель указывает на адрес памяти определенного типа данных. Функция может изменить содержимое по адресу, на который указывает указатель.

Изначально переменная x равна 1,7.

В функции f1 функция только передаёт значения, но не изменяет. Для работы с этими значениями и только в области функции объявляются переменные, поэтому их изменение внутри функции после выполнения не изменяет значение переменной x.

После функции f1 значение переменной x не изменяется и равна 1,7.

Обращение по указателю позволяет хранить адрес переменной. Указатель содержит адрес памяти. Когда значение указателя изменяется, это влияет на адрес переменной, к которой он указывает.

В функции f2 функция изменяет содержимое по адресу, на который указывает указатель.

После функции f2 значение переменной x будет равна 1,8.

В функции f3 функция ссылается на существующую переменную. Когда значение ссылки изменяется, то значение исходной переменной также меняется. Ссылка выступает как альтернативное имя для переменной.

После функции f3 значение переменной x будет равна 1,9.