**МИНОБРНАУКИ РОССИИ**

**САНКТ-ПЕТЕРБУРГСКИЙ ГОСУДАРСТВЕННЫЙ**

**ЭЛЕКТРОТЕХНИЧЕСКИЙ УНИВЕРСИТЕТ**

**«ЛЭТИ» ИМ. В.И. УЛЬЯНОВА (ЛЕНИНА)**

**Кафедра МО ЭВМ**

**ОТЧЕТ**

**по лабораторной работе №0**

**по дисциплине «Программирование»**

**Тема: Название лабораторной работы из методички.** Пример для работы #4 **Функции. Передача аргументов по ссылке и по значению**

|  |  |  |
| --- | --- | --- |
| Студент(ка) гр. 0000 |  | Иванов И.И. |
| Преподаватель |  | Шолохова О.М. |

Санкт-Петербург

2015

**Цель работы.**

Формулировка цели работы из методички (Раздел “ЦЕЛЬ РАБОТЫ”)

Пример для работы #4

Ознакомление с парадигмой процедурного программирования. Получение

навыков реализации функций на языке программирования C++. Изучение различных способов передачи аргументов в функцию.

**Постановка задачи.**

Формулировка задания из методички (Раздел “ЗАДАНИЯ К ЛАБОРАТОРНОЙ РАБОТЕ”) + формулировка задания к варианту

Пример для работы #4, вариант 3-0

Разработать программу для выполнения индивидуального задания с использованием функций языка С++. Первая цифра шифра задания определяет номер одной из предыдущих лабораторных работ, а вторая – номер варианта задачи соответствующей лабораторной работы.

Задание:

Вычислить частичную сумму сходящегося бесконечного ряда разложения функции. (!Ошибка в методичке - задание к лабораторной 3 указано не полностью, формулировка есть в примере выполнения)

Рассматриваются ряды *f(x) = Σi = 0...∞ ui.* Для каждого индивидуального

задания определены вид элемента ряда *ui*, функция *f(x)*, область сходимости *D*,

если *D ≠ R*.

Вариант 1. *u(i) = (–1)ix2i+1 / (2i+1)!; f(x) = sin(x).*

**Основные теоретические положения.**

Описание основных решений, принятых при выполнении лабораторной работы

Например

для вычисления того то был использован такой то алгоритм

для хранения того то был выбран такой то тип данных

задачу можно разбить на N частей и далее описание каждой части

все дополнительные вычисления, которые не отражены в программе

За основу можно взять раздел “ТЕОРЕТИЧЕСКИЙ МАТЕРИАЛ” из методички.

Пример для работы #4

Найдем рекуррентное соотношение для вычисления очередного элемента

ряда:

Получилось линейное рекуррентное соотношение первого порядка:

, где . Первый элемент последовательности.

Вычисления суммы ряда построим по схеме итерации:

x – переменные *<i, ui, si>*, где *i* - номер итерации, *ui* = элемент ряда и *si =*

*= Σj = 0...iuj*;

x0 – начальные значения *<i, ui, si> = <0,x,x>*;

F(x) = F(*<i, ui, si>*) = *<i+1, ui×p(i), si + ui>*;

B(x) = B(*<i, ui, si>*) = *i < N*.

По условию . Однако ясно, что вычисления не могут продолжаться бесконечно. Так как ряд сходящийся, то . Так как для хранения суммы и текущего элемента ряда будем использовать тип с плавающей точкой, то при некотором *k* разность порядков текущего значения суммы sk и элемента ряда uk будет достаточно велика, чтобы при сложении *ui>k* не влияли на значение суммы. Для определения k будем использовать машинное эпсилон, определяемое по формуле *eps = min {e: (e + 1) > 1}*. Так как ряд знакочередующийся, то условием завершения итераций будет условие

B(x) = B(*<i, ui, si>*) = .

**Спецификация программы.**

*Назначение программы*. (*Программа предназначена для…)*

Пример для работы #4, вариант 3-0

Программа предназначена для вычисления синуса угла с помощью вычисления частичной суммы сходящегося бесконечного ряда разложения функции sin(x).

*Описание программы*. (Программа написана на языке …с использованием компилятора …Входные и выходные данные)

Пример для работы #4

Программа написана на языке C++ с использованием компилятора gcc 4.8.2. Входными данными для программы является значение угла в радианах, вводимое пользователем с клавиатуры. Проверка значений, вводимых пользователем отсутствует, поведение программы при некорректном вводе не определено. Выходными данными являются:

1. Значение синуса угла, полученное с помощью вычисления частичной сходящегося бесконечного ряда разложения функции (а также все элементы ряда и промежуточные значения суммы ряда);

2. Значение машинного эпсилон, используемого при вычислениях;

3. Значение синуса угла, полученное с помощью стандартной библиотеки C++;

4. Анализ полученных результатов (относительная и абсолютная погрешности).

*Пример диалога с пользователем*. (текст или скриншот(ы))

Машинное эпсилон = 2.22045e-16

Введите x: 5

x = 5.0000000000e+00

Cлагаемое u( 1) = -2.0833333333e+01 сумма s( 1) = -1.5833333333e+01

Cлагаемое u( 2) = +2.6041666667e+01 сумма s( 2) = +1.0208333333e+01

Cлагаемое u( 3) = -1.5500992063e+01 сумма s( 3) = -5.2926587302e+00

Cлагаемое u( 4) = +5.3822889109e+00 сумма s( 4) = +8.9630180776e-02

Cлагаемое u( 5) = -1.2232474798e+00 сумма s( 5) = -1.1336172990e+00

Cлагаемое u( 6) = +1.9603324996e-01 сумма s( 6) = -9.3758404902e-01

Cлагаемое u( 7) = -2.3337291662e-02 сумма s( 7) = -9.6092134068e-01

Cлагаемое u( 8) = +2.1449716601e-03 сумма s( 8) = -9.5877636902e-01

Cлагаемое u( 9) = -1.5679617398e-04 сумма s( 9) = -9.5893316520e-01

Cлагаемое u(10) = +9.3331055943e-06 сумма s(10) = -9.5892383209e-01

Cлагаемое u(11) = -4.6112181790e-07 сумма s(11) = -9.5892429321e-01

Cлагаемое u(12) = +1.9213409079e-08 сумма s(12) = -9.5892427400e-01

Cлагаемое u(13) = -6.8423821507e-10 сумма s(13) = -9.5892427468e-01

Cлагаемое u(14) = +2.1066447508e-11 сумма s(14) = -9.5892427466e-01

Cлагаемое u(15) = -5.6630235238e-13 сумма s(15) = -9.5892427466e-01

Cлагаемое u(16) = +1.3406779176e-14 сумма s(16) = -9.5892427466e-01

Cлагаемое u(17) = -2.8165502470e-16 сумма s(17) = -9.5892427466e-01

Cлагаемое u(18) = +5.2863180311e-18 сумма s(18) = -9.5892427466e-01

S(x) = -9.5892427466e-01 sin x = -9.5892427466e-01 n = 18

Абсолютная погрешность = +1.1102230246e-16

*Реализация*. (описание всех функций и их спецификаций, глобальных переменных (желательно, чтобы их не было))

Пример

Глобальные переменные: (в коде их нет, используется как пример)

1. bool isOverflow - флаг переполнения, если значение true, то вычисления прервались из-за переполнения.

Функции:

1. uint computeSum(uint n, uint m, uint& i, uint& a, bool &isOverflow);

Вычисление суммы sum (i=0...n) (m!/(m-i)!)

Входные параметры:

значения m и n (коэффициенты в формуле)

Выходные параметры:

i - номер последнего слагаемого

a - значение последнего слагаемого

isOverflow - флаг переполнения (если isOverflow true то вычисления прервались из-за переполнения суммы)

Возвращаемое значение: сумма ряда sum (i=0...n) (m!/(m-i)!)

Предусловие: m >= n >= 0

Постусловие:

Если переполнения не произошло: возвращаемое значение sum (i=0...n) (m!/(m-i)!), i = n, a = (m!/(m-n)!), isOverflow = false

Если на шаге k+1 произошло переполнение: функция вернула значение sum (i=0...k) (m!/(m-i)!), i = k, a = (m!/(m-k)!), isOverflow = true

Побочные действия (в коде их нет, используется как пример):

В случае переполнения меняет значение глобальной переменной isOverflow - устанавливает значение true, если вычисления прервались из-за переполнения.

Зависимости от глобальных переменных (в коде их нет, используется как пример):

Если значение переменной x равно 0, то … (Лучше не использовать глобальные неконстантные переменные!)

**Тестирование.**

Тут специфично для 4й работы, по вашему усмотрению в таблице должны быть показательные значения

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Входные данные  x | Тип данных, используемый при вычислениях | Значение sin(x), полученный суммированием ряда | Значение sin(x), полученный с помощью стандартной библиотеки C++ | Абсолютная погрешность | Относительная погрешность |
| 1.0 | float | ... | ... | ... | ... |
| 1.0 | double |  |  |  |  |
|  |  |  |  |  |  |

**Выводы.**

Согласуются с целями

В ходе выполнения лабораторной работы были получены навыки реализации функций на языке программирования C++ и изучены различные способы передачи аргументов в функцию.

(Желательно перечислить список новых навыков, которые были получены, описать выводы по полученным результатам (например Как и следовало ожидать, погрешность вычислений при использовании long long double меньше. чем при использовании float))

**Приложение А. Исходный код.**

(Обратите внимание на шапку, теперь мы будем требовать ее всегда)

//

// Лабораторная работа №4

// lab4.cpp

// Иванов Иван, группа 6382

// 14.09.16

//

// Программа для вычисления синуса угла

(Пример кода для другого варианта)

#include <climits>

#include <iostream>

using namespace std;

using uint = unsigned int;

// Вычисление суммы sum (i=0...n) (m!/(m-i)!);

// Предусловие: m >= n >= 0

// Постусловие:

// i - номер последнего слагаемого

// a - значение последнего слагаемого

// isOverflow - флаг переполнения (если isOverflow true то

// вычисления прервались из-за переполнения суммы)

// Возвращает значение вычисленной суммы

uint computeSum(uint n, uint m, uint& i, uint& a, bool &isOverflow);

int main()

{

uint m, n;

cout << "Вычисление суммы s = sum(i = 0...n) (m!/ (m - i)!), m >= n >= 0" << endl;

cout << "Введите n > 0: ";

cin >> n;

cout << "n = " << n << endl;

cout << "Введите m > " << n << ": ";

cin >> m;

cout << "m = " << m << endl;

bool isOverflow; // флаг переполнения

uint a, i;

uint s = computeSum(n, m, i , a, isOverflow);

cout << "При i = " << i << ", последнее слагаемое a = " << a << ", сумма s = " << s << endl;

if (isOverflow)

cout << "Ошибка: При i = " << i + 1 << " произойдет переполнение." << endl;

}

uint computeSum(uint n, uint m, uint& i, uint& a, bool &isOverflow)

{

a = 1; // текущее значение a

i = 0; // номер шага

isOverflow = false; // флаг переполнение

uint s = 1; // текущая сумма последовательности

// Инвариант цикла: a=a(i) && s = sum (j=0..i) (a(j))

while ((i < n) && !isOverflow)

{

cout << "i = " << i << ", a = " << a << ", s = " << s << endl;

a = (m - i) \* a;

s = s + a;

i = i + 1;

if ((i < n) && ((UINT\_MAX - s) / (m - i) < a))

isOverflow = true;

}

return s;

}

**Приложение Б. Блок-схема.**

**(Источник https://ru-static.z-dn.net/files/da7/47ea3317011de792dd5234e2f10434c7.jpg)**

