|  |  |
| --- | --- |
| **Gerb-BMSTU_01** | **Министерство науки и высшего образования Российской Федерации**  Калужский филиал  федерального государственного бюджетного  образовательного учреждения высшего образования  ***«Московский государственный технический университет имени Н.Э. Баумана (национальный исследовательский университет)»***  ***(КФ МГТУ им. Н.Э. Баумана)*** |

**ФАКУЛЬТЕТ** ***«Информатика и управление»\_\_\_\_\_\_\_\_\_\_\_\_***

**КАФЕДРА** ***«Системы автоматического управления»***

**ЛАБОРАТОРНАЯ РАБОТА №2**

**ДИСЦИПЛИНА: «Язык программирования Си»**

**ТЕМА: «Операторы языка С. Управляющие конструкции и инструкции С»**   
  
  
  
Выполнил: студент гр. ИУК1-22Б \_\_Логинов А.А.\_\_  
  
Проверил: \_\_Акименко Д.А. \_\_   
  
  
  
Дата сдачи (защиты) лабораторной работы:

Результаты сдачи (защиты):

Количество рейтинговых баллов:  
  
  
Оценка:

Калуга, 2024

**Цель работы:**

1. Знакомство студентов с операторами С, правилами приоритета, ассоциативности и лексического разбора выражений;

2. Знакомство студентов с управляющими инструкциями и конструкциями С;

3. Написание и отладка простых программ, реализующих несложные алгоритмы вычислительной математики.

**Задание:** вычислить m-значений заданной функции f(x) на интервале [a,b].

**Псевдокод:**

ЦИКЛ ДЛЯ x от x\_начальное до x\_конечное с шагом step

ВЫЧИСЛИТЬ

Значение функции в точке с помощью математической библиотеки;

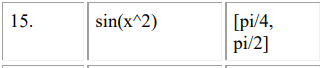
Значение функции в точке итерационным методом;

Оценить точность вычислений;

КОНЕЦ ЦИКЛА

Вывести результаты вычислений

Вариант 15



**Листинг программы:**

#define \_CRT\_SECURE\_NO\_WARNINGS

#include <stdio.h>

#include <math.h>

#define REAL double

#define type(x) t(x)

#define t(z) #z

#define Kmax 10000

#define M\_PI 3.14159265358979323846

int main()

{

REAL xmin = M\_PI / 4;

REAL xmax = M\_PI / 2;

int N = 10;

REAL dx = (xmax - xmin) / (N - 1);

for (REAL x = xmin; x < xmax + dx; x += dx) {

REAL e, e1, S;

int k = 0;

e = x;

do {

k++;

e /= 2.0;

e1 = x = e;

} while (e1 > x && k < Kmax);

k = 0;

REAL a = x;

S = 0;

REAL h = 0;

while (fabs(a) > e) {

k += 1;

h = -x \* x / (2 \* k \* (2 \* k + 1));

a \*= h;

S += a;

}

S /= x;

REAL f2 = sin(x \* x);

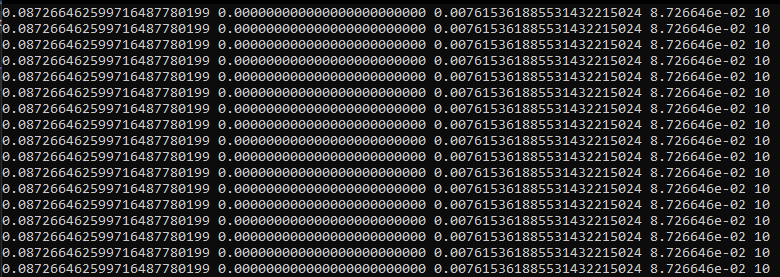
printf("%.24f %.24f %.24f %e %d\n", x, S, f2, e, N);

}

return 0;

}

**Результат работы кода:**



**Ответ на контрольные вопросы:**

1. Какие типы операторов есть в С?

Операторы бывают различных типов, таких как арифметические, логические, операторы сравнения, операторы присваивания, операторы условного выполнения и циклические операторы

1. Что такое правило ассоциативности? Где оно используется?

Ассоциативность – в каком направлении, будут выполнять операторы одного выражения.

3. Сколько рангов имеют операторы С? Операторы каких рангов имеют

правую ассоциативность?

Всего в языке С - 15 рангов операторов и два направления ассоциативности - левое и правое. Операторы ранга 1 имеют наивысший приоритет и выполняются первыми, оператор 15 ранга – последним. Правую ассоциативность имеют операторы 2-го, 13-го и 14-го рангов – т. е. унарные операторы, условный тернарный оператор и группа операторов присваивания.

4. Где используются ранги операторов С?

Операторы ранга 1 имеют наивысший приоритет и выполняются первыми, оператор 15 ранга – последним. Если выражение содержит операторы одного ранга, то порядок выполнения определяется в соответствии с правилом ассоциативности - либо слева направо (->), либо справа налево (<-).

5. Чем логические операторы отличаются от операторов отношений?

Операторы отношения применимы для вычисления соотношений между операндами. Логические операторы, используя правила формальной логики, также возвращают соотношения между операндами.

6. Чем отличаются постфиксные и префиксные формы операторов декремента и инкремента?

Декремента и инкремента имеют две формы:

**Префиксную**, например ++i+j, - оператор - увеличение значения операнда на 1 до его использования в выражении. Ассоциативность справа в соответствии со стандартом;

**Постфиксную** – например i++-j, оператор - увеличение значения операнда на 1 после его использования в выражении, ассоциативность слева в соответствии со стандартом.

7. Можно ли и для чего использовать операторы сдвига в арифметических

выражениях С? Приведите пример.

Побитовые операции могут использоваться для оптимизации некоторых алгоритмов. Например, операции сдвига могут быть использованы для умножения или деления числа на два, что может быть полезно, например, при работе с большими массивами данных

8. Что такое условный тернарный оператор? Как можно на его основе определить: 1. -абсолютную величину вещественного аргумента? 2.- минимальное или максимальное значение из пары аргументов?

В отличие от унарных и бинарных операторов условный тернарный оператор используется с тремя операндами. В изображении оператора применяются два символа '?' и ':' и три выражения-операнда: выражвние\_1 ? выражение\_ 2 : выражение\_3. Первым вычисляется значение выражения\_1. Если оно истинно, - не равно нулю, то вычисляется значение выражения\_2, которое становится результатом. Если при вычислении выражения\_1 получится 0, то в качестве результата берется значение выражения\_3.

**Пример:**

х < 0 ? -х : х - выражение возвращает абсолютную величину переменной х, или

x <= y ? x:y - выражение возвращает минимум из двух значений.

9. Какие скобки выступают в С в качестве операторов? Приведите примеры.

Круглые скобки ( ) в С могут быть унарным оператором – (OP) – где ОР – выражение или часть выражения, в котором операторы становятся операторами первого ранга, по отношению к операторам вне круглых скобок. Например, выражение i+j\*k будет выполняться так: 1. j\*k 2. i+j\*k. Выражение (i+j)\*k будет выполняться так: 1. i+ j 2. ( i+j)\*k. Порядок выполнения операторов в круглых скобках будет определяться приоритетами операторов и их ассоциативностью. Круглые скобки являются бинарным оператором при обращении к функции: имя\_функции(список\_аргументов), где операндами служат имя\_функции и список\_аргументов.

В выражении имя\_массива[индекс] операндами для оператора [ ] служат имя\_массива и индекс.

10. Что такое оператор явного преобразования типа? Когда надо его использовать? Пример.

Явное приведение (изменение) типа — это специальная конструкция языка программирования, которая указывает, как трактовать тип определенной переменной (или результат выражения). Приведение типов игнорирует любую дополнительную информацию (и никогда ее не добавляет).

11. Каковы правила выделения лексем?

Когда компилятор выделяет отдельную лексему, он последовательно объединяет столько символов, сколько возможно, прежде чем перейти к обработке следующей лексемы. Поэтому лексемы, не разделенные пробельными символами, могут быть проинтерпретированы неверно.

12.Какие операторы относятся к управляющим инструкциям С?

Это операторы условного и безусловного переходов, операторы

управления циклами, переключателями, функциями.

13. Какие операторы относятся к управляющим конструкциям С?

Это операторы циклов, оператор – переключатель.

14. В чем отличие оператора цикла с постусловием С от аналогичного оператора ПАСКАЛЬ?

Различие заключается в том, что цикл while проверяет условие до выполнения действий, в то время как repeat проверяет условие после выполнения действий. Это гарантирует хотя бы одно выполнение действий до завершения цикла.

**Вывод:** Ознакомился с операторами С, правилами приоритетов, ассоциативности и лексического разбора выражений. Ознакомился с операторами управляющими конструкции и инструкции. Написал программу, реализующую не сложные алгоритмы вычислений математики.