**МИНИСТЕРСТВО ТРАНСПОРТА РОССИЙСКОЙ ФЕДЕРАЦИИ**

**федеральное государственное автономное образовательное**

**учреждение высшего образования**

**«РОССИЙСКИЙ УНИВЕРСИТЕТ ТРАНСПОРТА»**

**(РУТ(МИИТ)**

Кафедра «Вычислительные системы, сети и информационная безопасность»

**Отчет По Дисциплине**

**«Алгоритмизация и программирование»**

**Лабораторная работа №3**

*Направление:* 10.03.01*Информационная безопасность*

*Профиль:**Безопасность компьютерных систем*

Выполнил:  
студент группы УИБ-112

Орлов Андрей Витальевич

Проверил:

Старший преподаватель Никольская Марина Николаевна

(должность, ФИО)

Старший преподаватель Цыганова Наталия Алексеевна

(должность, ФИО)

Москва 2021 г.

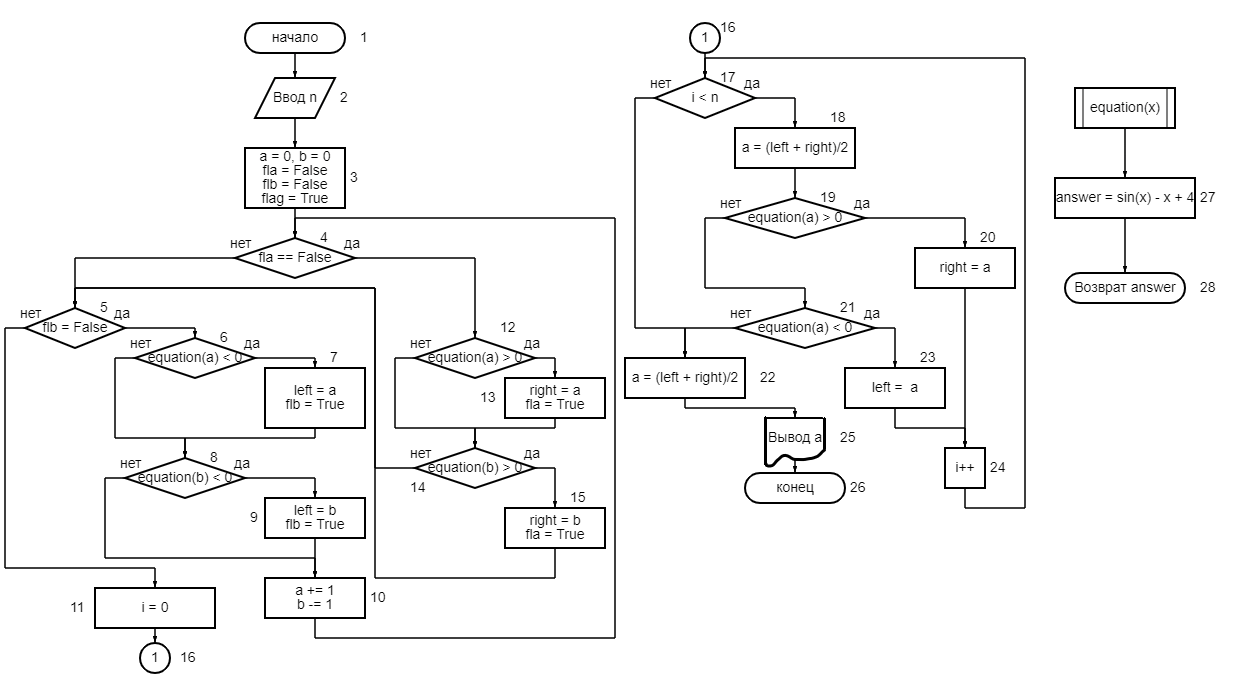
**Задание №3**

Задание 3: Вычислить корни уравнения sin(x) – x + 2 = 0 методом деления отрезка пополам.

1. Таблица имён:

|  |  |  |
| --- | --- | --- |
| Исходные данные | | |
| n | целочисленный | Переменная точности вычисления ответа, именно столько раз отрезок ответа будет приближен к решению уравнения |
| Рабочие переменные | | |
| a | вещественный | Изначально переменная с увеличивающимся положительным значением, используется для поиска первейшего отрицательного или положительного значения уравнения, далее используется для подбора более точного нулевого значения уравнения |
| b | целочисленный | Изначально переменная с увеличивающимся отрицательным значением, используется для поиска первейшего отрицательного или положительного значения уравнения |
| fla | булевый | Флаг, сигнализирующий о факте нахождения отрицательного значения функции |
| flb | булевый | Флаг, сигнализирующий о факте нахождения положительного значения функции |
| left | вещественный | Левая граница отрезка ответа |
| right | вещественный | Правая граница отрезка ответа |
| Результат | | |
| a | вещественный | Приближённый ответ уравнения |

1. Блок-схема:



1. Отладочный пример:

Вариант I

1. Начало  
2. Ввод с клавиатуры n = 4  
3.Объявление переменных  
а = 0, b = 0, fla = False, flb = False, flag = True  
4. fla == False, да, идём к блоку 12  
12. Идём в блок 27 для вычисления подпрограммы  
 27. answer = sin(0) – 0 + 4 = 4  
 28. Возврат 4, конец подпрограммы  
4 > 0, да, идём в блок 13  
13.Переприсваивание переменных right = 0, fla = True  
14. Идём в блок 27 для вычисления подпрограммы  
 27. answer = sin(0) – 0 + 4 = 4  
 28. Возврат 4, конец подпрограммы  
4 > 0, да, идём в блок 15  
15. right = 0, fla = True, идём в блок 5  
5. flb == False, да, переход в блок 6  
6. Идём в блок 27 для вычисления подпрограммы  
 27. answer = sin(0) – 0 + 4 = 4  
 28. Возврат 4, конец подпрограммы  
4 < 0, нет, идём в блок 8  
8. Идём в блок 27 для вычисления подпрограммы

27. answer = sin(0) – 0 + 4 = 4  
 28. Возврат 4, конец подпрограммы

4 < 0, нет, идём в блок 10  
10. Переприсваивание переменных  
а = а + 1 = 0 + 1 = 1, b = b = b – 1 = 0 – 1 = -1.  
4. fla == False, нет, идём в блок 5.  
5. flb == False, да, идём в блок 6.

6. Идём в блок 27 для вычисления подпрограммы

27. answer = sin(1) – 1 + 4 = 0.84 + 3 = 3.84  
 28. Возврат 3.84, конец подпрограммы

3, 84 < 0, нет, идём в блок 8.  
8. Идём в блок 27 для вычисления подпрограммы

27. answer = sin(-1) + 1 + 4 = -0.84 + 5 = 4.16  
 28. Возврат 4.16, конец подпрограммы

4.16 < 0, нет, идём в блок 10.  
10. Переприсваивание переменных  
a = a + 1 = 1 + 1 = 2, b = b – 1, b = -1 -1 = -2.  
4. fla == False, нет, идём в блок 5  
5. flb == False, да, идём в блок 6

6. Идём в блок 27 для вычисления подпрограммы

27. answer = sin(2) – 2 + 4 = 0.9 + 2 = 2.9  
 28. Возврат 2.9, конец подпрограммы  
2.9 < 0, нет, идём в блок номер 8.  
8. Идём в блок 27 для вычисления подпрограммы

27. answer = sin(-2) + 2 + 4 = -0.9 + 6 = 5.1  
 28. Возврат 5.1, конец подпрограммы

5.1 < 0, нет, идём в блок номер 10  
10. Переприсваивание переменных  
a = a + 1 = 2 + 1 = 3, b = b – 1 = - 2 - 1 = -3

4. fla == False, нет, идём в блок 5  
5. flb == False, да, идём в блок 6.

6. Идём в блок 27 для вычисления подпрограммы

27. answer = sin(3) – 3 + 4 = 0.14 + 1 = 1.14  
 28. Возврат 1.14, конец подпрограммы  
1.14 < 0, нет, идём в блок номер 8.  
8. Идём в блок 27 для вычисления подпрограммы

27. answer = sin(-3) + 3 + 4 = -0.14 + 7 = 6.86  
 28. Возврат 6.86, конец подпрограммы

6.86 < 0, нет, идём в блок номер 10

10. Переприсваивание переменных  
a = a + 1 = 3 + 1 = 4, b = b – 1 = - 3 - 1 = - 4

4. fla == False, нет, идём в блок 5  
5. flb == False, да, идём в блок 6.   
6. Идём в блок 27 для вычисления подпрограммы

27. answer = sin(4) – 4 + 4 = -0.75  
 28. Возврат -0.75, конец подпрограммы  
-0.75 < 0, да, идём в блок номер 7.

7. left = a = 4, flb = True  
8. Идём в блок 27 для вычисления подпрограммы

27. answer = sin(-4) + 4 + 4 = 0.75 + 7 = 7.75  
 28. Возврат 7.75, конец подпрограммы

7.75 < 0, нет, идём в блок номер 10  
10. Переприсваивание переменных  
a = a + 1 = 4 + 1 = 5, b = b – 1 = -4 -1 = -5  
4. fla == False, нет, идём в блок 5.  
5. flb == False, нет, идём в блок 11.  
11. i = 0  
16. Ссылка на блок 17.  
17. 0 < 4, да, идём в блок номер 16.  
16. Переприсваивание переменной  
a = (4 + 0)/2 = 2  
19. Идём в блок 27 для вычисления подпрограммы  
 27. answer = sin(2) – 2 + 4 = 0.9 + 2 = 2.9  
 28. Вывод 2.9  
 29. Возврат 2.9, конец подпрограммы  
2.9 > 0, да, идём к блоку 20  
20. Переприсваивание переменных  
right = a = 2  
24. i = i + 1 = 0 + 1 = 1  
17. 1 < 4, да, идём в блоку 18  
18. a = (4 + 2)/2 = 3  
19. Идём в блок 27 для вычисления подпрограммы  
 27. answer = sin(3) – 3 + 4 = 0.14 + 1 = 1.14  
 28. Возврат 1.14, конец подпрограммы  
1.14 > 0, да, идём к блоку 20.  
20. Переприсваивание переменных.   
right = 3  
24. i = i + 1 = 1 + 1 = 2  
17. 2 < 4, да, идём к блоку 18  
18. a = (4 + 3)/2 = 3.5  
19. Идём в блок 27 для вычисления подпрограммы  
 27. answer = sin(3.5) – 3.5 + 4 = -0.35 + 0.5 = 0.15  
 28. Возврат 0.15, конец подпрограммы  
0.15 > 0, да, идём к блоку 20.  
20. Переприсваивание переменных  
right = a = 3.5  
24. i = i + 1 = 2 + 1 = 3  
17. 3 < 4, да, идём к блоку 18  
18. a = (4 + 3.5)/2 = 3.75

19. Идём в блок 27 для вычисления подпрограммы  
 27. answer = sin(3.75) – 3.75 + 4 = -0.57 + 0.25 = -0.32  
 28. Возврат -0.32, конец подпрограммы  
-0.32 > 0, нет, идём к блоку 21.  
21. Идём в блок 27 для вычисления подпрограммы  
 27. answer = sin(3.75) – 3.75 + 4 = -0.57 + 0.25 = -0.32  
 28. Возврат -0.32, конец подпрограммы  
-0.32 < 0, да, идём в блок 23  
23. Переприсваивание переменных  
left = a = 3.75  
24. I = i + 1 = 3 + 1 = 4  
17. 4 < 4, нет, идём в блок 22  
22. a = (3.75 + 3.5)/2 = (7.25)/2 = 3.625  
25. Вывод a  
26. Конец

1. Код программы:

#include <iostream>

#include <cmath>

#include <iomanip> // Подключение библиотеки для манипуляций вводом\выводом, в частности - setprecision().

using namespace std;

float equation(float x) {

return double(sin(x) - x + 4);

}

int main() {

setlocale(LC\_ALL, "Russian");

// Объявление переменных точностей  
 int n, accuracy;

cout << "Введите точность вычислений корня уравнения: ";

cin >> n;

cout << "Введите количество знаков после запятой (Незначащие нули не будут выведены): ";

cin >> accuracy;

double a = 0, b = 0, right, left;

// Объявление флагов нахождения границ отрезка функции в формате [-x; x], x > 0

bool flag = true, fla = false, flb = false;

while (flag) {

if (!fla) { // Пока не найдена правая граница отрезка

if (equation(a) > 0) {

right = a; // правая граница отрезка найдена

fla = true; // меняем булевое значение флага

}

if (equation(b) > 0) {

right = b; // правая граница отрезка найдена,

fla = true; // меняем булевое значение флага

}

}

if (!flb) { // Пока не найдена левая граница отрезка

if (equation(a) < 0) {

left = a; // левая граница отрезка найдена,

flb = true; // меняем булевое значение флага

}

if (equation(b) < 0) {

left = b; // левая граница отрезка найдена,

flb = true; // меняем булевое значение флага, }

}

a += 1;

b -= 1;

if (fla && flb) { // Если нашлись границы отрезка

break; // Выходим из цикла

}

}

// Инициализация цикла вычисления корня уравнения

for (int i = 0; i < n; i++) {

// Переменная а используется для краткости программы, в нее записываем середину действующего отрезка

a = (left + right) / 2;

// Если значение функции меньше нуля, тогда середина становится левой границей отрезка

if (equation(a) > 0) {

right = a;

}

// Если значение функции меньше нуля, тогда середина становится левой границей отрезка

else if (equation(a) < 0) {   
 left = a;

}

// Если границы отрезка идентичны друг другу, тогда мы нашли корень, выходим из цикла

else {

break;

}

}

a = (left + right) / 2;

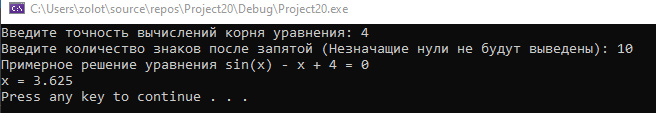
cout << "Примерное решение уравнения sin(x) - x + 4 = 0\nx = " << setprecision(accuracy) << a << endl;

system("pause");

return 0;

}

1. Результат выполнения работы программы:



Отладочный пример №1

1. Вывод:

В ходе выполнения работы были изучены способы нахождения решения уравнения путём нахождения приблизительного значения ответа методом исследования числовой оси на отрезок с границами, в которых функция различается по знаку. Далее с помощью деления отрезка пополам запрашиваемое количество раз находим приблизительный ответ.

Было проделано несколько опытов на других уравнениях, где такой способ вполне сработал, так что алгоритм универсален.

На контрольных примерах мы убедились, что алгоритм работает верно.

Был оформлен комплект документации на программный код.