МИНИСТЕРСТВО ОБРАЗОВАНИЯ И НАУКИ РОССИЙСКОЙ

ФЕДЕРАЦИИ

Федеральное государственное бюджетное образовательное

учреждение высшего образования

«Московский Авиационный Институт

(Национальный Исследовательский Университет)»

Институт №8 «Компьютерные науки и прикладная математика»

Кафедра 806 «Вычислительная математика и программирование»

**Курсовой проект**

по курсу “Фундаментальная информатика” 1 семестра

Задание 4. Процедуры и функции в качестве параметров.

Выполнил Епифанов Е. В., №9 по списку

Группа М8О-112Б-22

Проверил Никулин С. П., каф. 806

Москва, 2022

1. **Постановка задачи**:

Составить программу на языке Си с процедурами решения трансцендентных алгебраических уравнений различными численными методами (итераций, Ньютона и половинного деления - дихотомии). Нелинейные уравнения оформить как параметр-функции, разрешив относительно неизвестной величины в случае необходимости. Применить каждую процедуру к решению двух уравнений, заданных двумя строками таблицы.

**Вариант №17,18**

Изображение выглядит как стол

Автоматически созданное описаниеИзображение выглядит как стол

Автоматически созданное описание

1. **Метод решения:**

Создадим функции, которые будут возвращать значение функций и их производных из данных вариантов в точке x. Необходимо создать функции, которые получают на вход функцию, и границы отрезка, после чего решают заданное уравнение. После этого результаты выводятся в виде таблицы.

1. **Метод дихотомии:**

Изображение выглядит как текст

Автоматически созданное описание

1. **Метод итераций:**

Изображение выглядит как текст

Автоматически созданное описание

1. **Метод Ньютона:**

Изображение выглядит как текст

Автоматически созданное описание

1. **Сведения о программе**

Программа написана на языке Си с помощью редактора Emacs в OC Linux. Редактор текстов: GNU Emacs 28.2.

Компилятор gcc Linux.

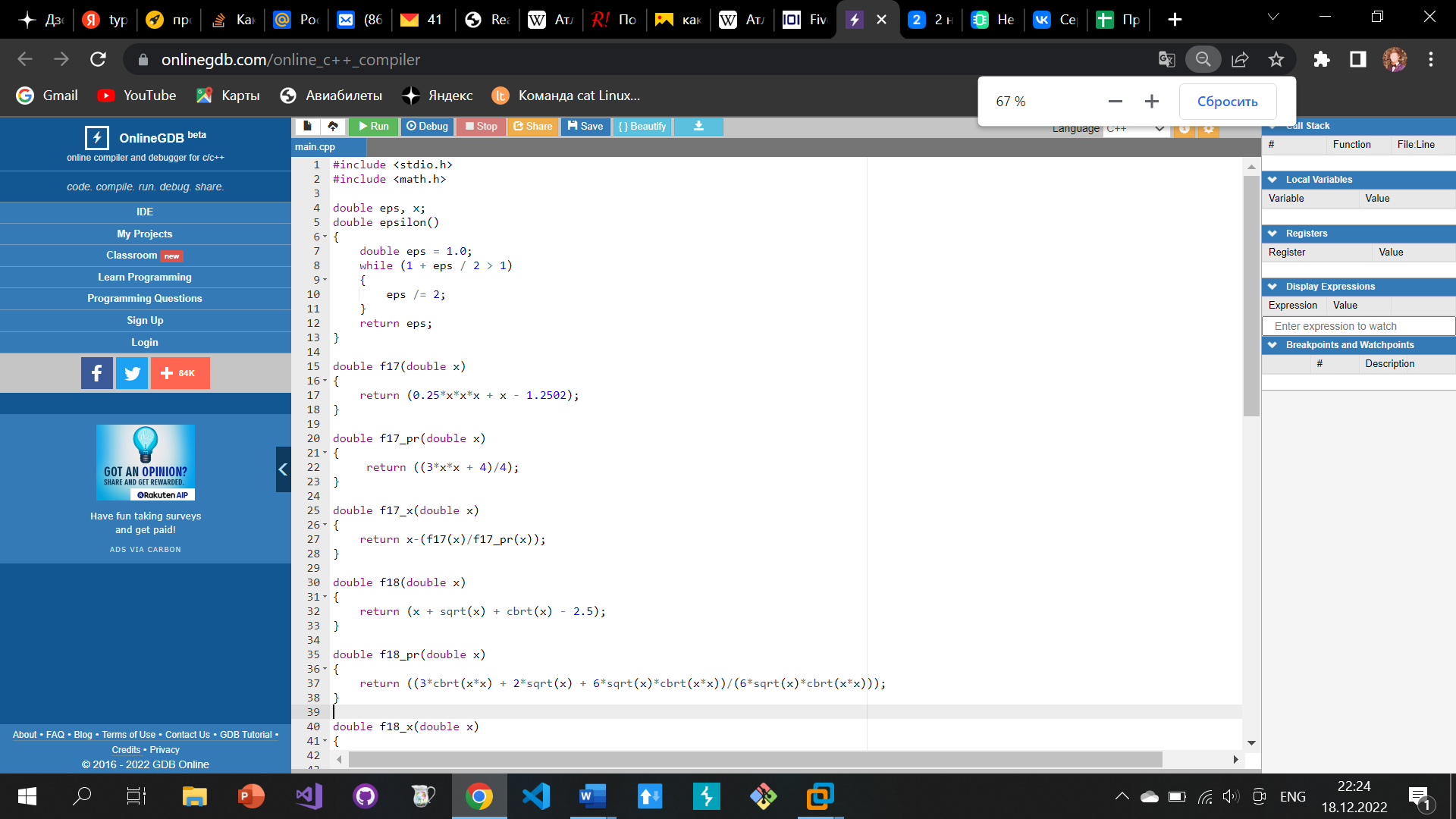
Операционная система семейства UNIX, наименование: Linux Ubuntu (22.04) 5.15.0-48-generic #54-Ubuntu SMP Fri Aug 26 13:26:29 UTC 2022 x86\_64 x86\_64 x86\_64 GNU/Linux.

Программа компилируется с помощью компилятора gcc.

1. **Функциональное назначение­­**

Программа предназначена для нахождения решений уравнений на определенном отрезке. Используются переменные типа double и long long. Ввод данных в программе отсутствует ввиду ненадобности, потому и отсутствуют всякие ограничения.

1. **Описание алгоритма**



Изображение выглядит как текст, снимок экрана, компьютер, ноутбук

Автоматически созданное описание

Изображение выглядит как текст, снимок экрана, монитор, компьютер

Автоматически созданное описание

**Программа в виде текста:**

#include <stdio.h>

#include <math.h>

double eps, x;

double epsilon()

{

double eps = 1.0;

while (1 + eps / 2 > 1)

{

eps /= 2;

}

return eps;

}

double f17(double x)

{

return (0.25\*x\*x\*x + x - 1.2502);

}

double f17\_pr(double x)

{

return ((3\*x\*x + 4)/4);

}

double f17\_x(double x)

{

return x-(f17(x)/f17\_pr(x));

}

double f18(double x)

{

return (x + sqrt(x) + cbrt(x) - 2.5);

}

double f18\_pr(double x)

{

return ((3\*cbrt(x\*x) + 2\*sqrt(x) + 6\*sqrt(x)\*cbrt(x\*x))/(6\*sqrt(x)\*cbrt(x\*x)));

}

double f18\_x(double x)

{

return x-(f18(x)/f18\_pr(x));

}

typedef double(\*f\_pointer)(double);

double dihot(f\_pointer f, double a, double b) // Метод дихотомии

{

double x;

while (fabs(a-b) >= eps)

{

x = (a+b) / 2;

if (f(a)\*f(x) >= 0)

a = x;

else

b = x;

}

return x;

}

double iter(f\_pointer f, double a, double b) // Метод итераций

{

double x = (a+b) / 2;

while (fabs(x-f(x)) >= eps)

{

x = f(x);

}

return x;

}

double newton(f\_pointer f, f\_pointer fd, double a, double b) // Метод Ньютона

{

double x = (a+b) / 2;

while (fabs(f(x)/fd(x)) >= eps)

{

x = x-(f(x)/fd(x));

}

return x;

}

int main()

{

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

printf("| Вариант 17,18 - Епифанов Евгений - М8О-112Б-22 |\n");

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

long long k = 5;

eps = epsilon() \* k;

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

printf("| № Уравнения | Отрезок | Метод дихотомии | Метод итераций | Метод Ньютона |\n");

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

printf("| 17 | [0.0, 2.0] | %.20f | %.20f | %.20f |\n", dihot(f17, 0.0, 2.0),iter(f17\_x, 0.0, 2.0), newton(f17, f17\_pr, 0.0, 2.0));

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

printf("| 18 | [0.4, 1.0] | %.20f | %.20f | %.20f |\n", dihot(f18, 0.4, 1.0), iter(f18\_x, 0.4, 1.0), newton(f18, f18\_pr, 0.4, 1.0));

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

}

**Алгоритм:**

1) Подсчёт машинного эпсилон (eps) через функцию epsilon().

2) Подсчёт функции 17 варианта.

3) Подсчёт производной функции 17 варианта.

4) Подсчёт выражения x = f(x), где f(x) = функция 17 варианта.

5) Подсчёт функции 18 варианта.

6) Подсчёт производной функции 18 варианта.

7) Подсчёт выражения x = f(x), где f(x) = функция 18 варианта.

8) Функция “Метод дихотомии”.

9) Функция “Метод итераций”.

10) Функция “Метод Ньютона”.

11) Вывод таблицы с данными.

1. **Описание переменных**

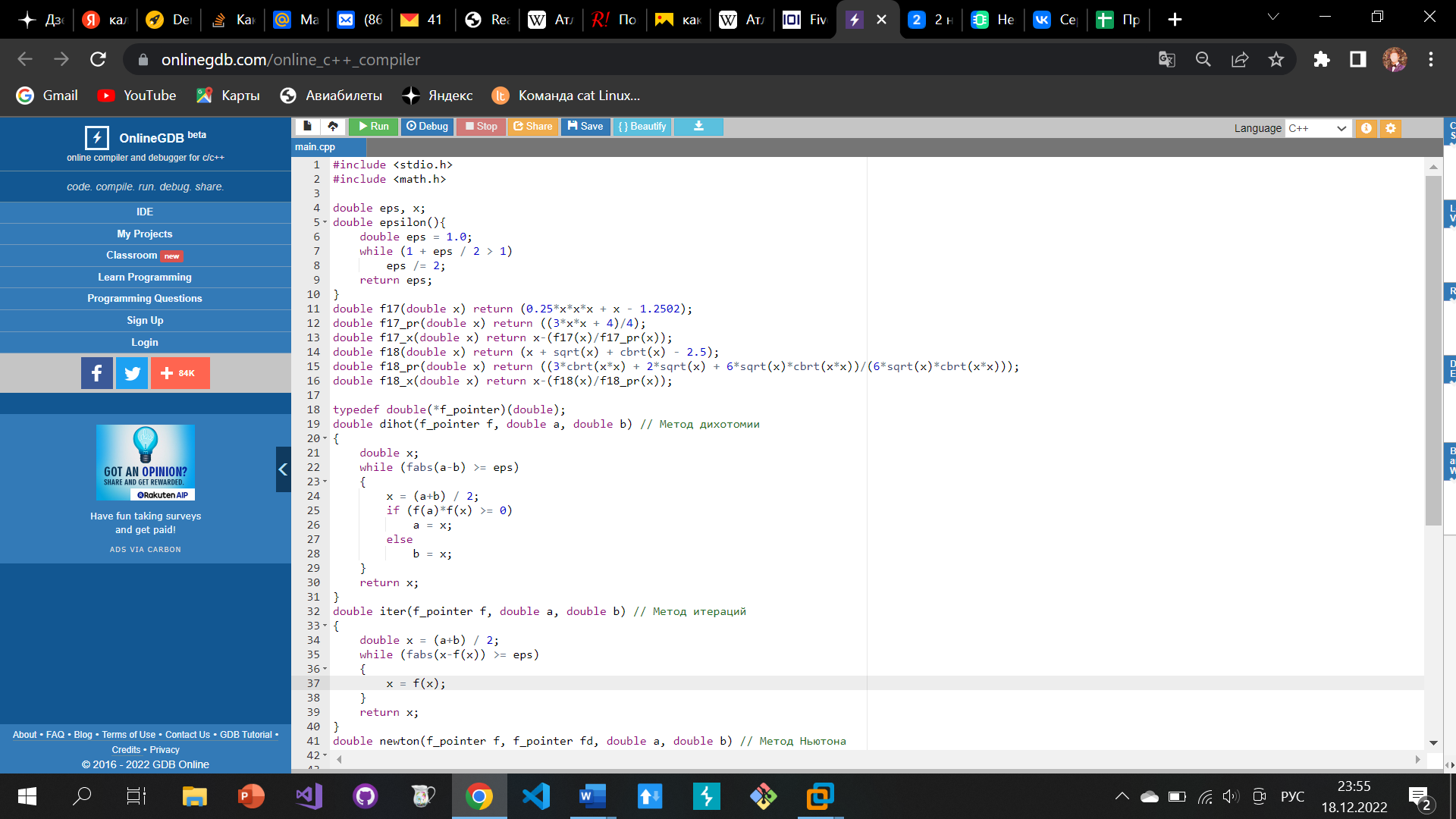
|  |  |
| --- | --- |
| Константы | Значение |
| a (double) | 0.0 / 0.4 |
| b (double) | 2.0 / 1.0 |
| k (long long) | 5 |

a и b – границы отрезка (1 отрезок / 2 отрезок), с которого берется x. k – константа для корректировки ограничивающего значения эпсилон.

|  |  |
| --- | --- |
| Переменная | Описание |
| eps (double) | Начальное значение эпсилон |
| x (double) | Искомое решение уравнения |
| f (f\_pointer) | Функция, используемая в другой функции |

1. **Подпрограммы**

В программе реализованы несколько функций: для подсчета значения eps, для подсчета первой основной функции, для подсчета её производной, для подсчета x = f(x) для первого уравнения, для подсчета второй основной функции, для подсчета её производной, для подсчета x = f(x) для второго уравнения, для подсчета значений методом дихотомии, итераций и Ньютона.



Изображение выглядит как текст, снимок экрана, монитор, внутренний

Автоматически созданное описание

1. **Входные данные**

Входных данных нет. Необходимые константы и уравнения заданы в программе изначально.

1. **Выходные данные**

В начале программы выводится плашка с данными о работе и её авторе. Затем поэтапно выводится таблица: шапка, строки с данными, дно. Выводится строчный текст и значения некоторых переменных и результатов функций, объявленных выше в программе.

1. **Тесты**

evgeniy2@evgeniy2:~$ emacs krs4.c

evgeniy2@evgeniy2:~$ cc krs4.c -lm

evgeniy2@evgeniy2:~$ ./a.out

--------------------------------------------------

| Вариант 17,18 - Епифанов Евгений - М8О-112Б-22 |

--------------------------------------------------

---------------------------------------------------------------------------------------------------------

| № Уравнения | Отрезок | Метод дихотомии | Метод итераций | Метод Ньютона |

|---------------|------------|------------------------|------------------------|------------------------|

| 17 | [0.0, 2.0] | 1.00011428011695269191 | 1.00011428011695313600 | 1.00011428011695313600 |

|---------------|------------|------------------------|------------------------|------------------------|

| 18 | [0.4, 1.0] | 0.73761924627746322525 | 0.73761924627746278116 | 0.73761924627746278116 |

Изображение выглядит как текст, снимок экрана, монитор, внутренний

Автоматически созданное описание

1. **Дневник отладки**

Работа выполнена 18.12.2022 в период с 19:26 до 23:12. Критических ошибок не возникало.

Многие ошибки, связанные с неточностью работы программы, легко заметить, так как в таблице должны выводиться относительно похожие на заранее известные числа значения – если присутствует ошибка, данные в строках будут сильно различаться. Для определения проблемы удобно использовать отслеживание значений переменных – так можно понять, на каком этапе программа выдала неверный ответ, и исправить проблему.

1. **Заключение**

В процессе выполнения данного задания я улучшил навыки вычисления и дальнейшего использования «машинного эпсилон». Я научился составлять программы на языке СИ, решающие уравнения методом итераций, дихотомии и Ньютона. Полученные значения являются корнями уравнений на изначально заданных отрезках. Они равны заявленным изначально, соответственно, программа работает корректно. Также я улучшил умение визуализировать выходные данные программы (в данном случае – вывод данных через таблицу).