Министерство науки и высшего образования РФ

ФГАОУ ВО Пермский национальный исследовательский

политехнический университет

Кафедра «Вычислительная математика, механика и биомеханика»

Отчет по лабораторной работе № 1

тема «Знакомство с языком программирования C»

по дисциплине «Языки программирования и методы трансляции»

Выполнил: студент группы ИСТ-22-1Б Синьковский Г.С.

Проверил: Батин С.Е.

Пермь, 2022

Содержимое

[Упражнение 1 3](#_Toc114935095)

[1.1. Постановка задачи 3](#_Toc114935096)

[1.2. Код программы 3](#_Toc114935097)

[1.3 Пример работы программы 3](#_Toc114935098)

[Упражнение 2 3](#_Toc114935099)

[2.1. Постановка задачи 3](#_Toc114935100)

[2.2. Код программы 4](#_Toc114935101)

[2.3. Примеры работы программы 4](#_Toc114935102)

[Упражнение 3 6](#_Toc114935103)

[3.1. Постановка задачи 6](#_Toc114935104)

[3.2 Код программы 6](#_Toc114935105)

[3.3. Пример работы программы 6](#_Toc114935106)

[Упражнение 4 6](#_Toc114935107)

[4.1. Постановка задачи 6](#_Toc114935108)

[4.2. Код программы 7](#_Toc114935109)

[4.3. Пример работы программы 7](#_Toc114935110)

[Упражнение 5 8](#_Toc114935111)

[5.1 Постановка задачи 8](#_Toc114935112)

[5.2 Код программы 8](#_Toc114935113)

[5.3 Пример работы программы 9](#_Toc114935114)

# Упражнение 1

## 1.1. Постановка задачи

Написать и запустить программу, выводящую строку “Hello, World!” в терминал на языке программирования C.

## 1.2. Код программы

#include <stdio.h>

int main (void)

{

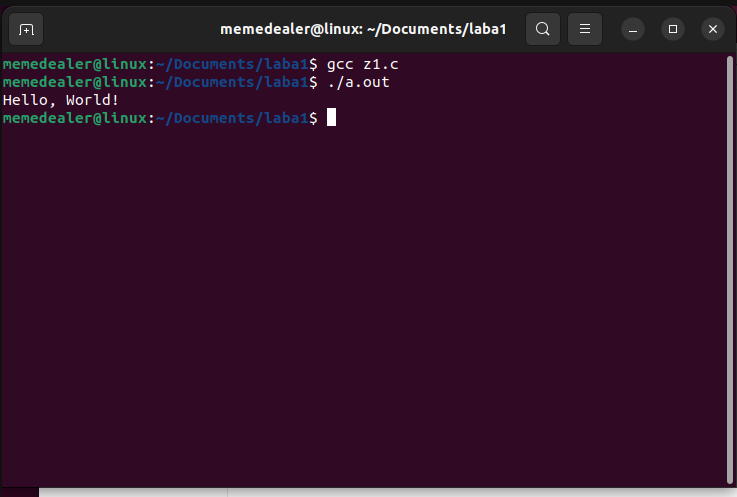
puts ("Hello, World!"); // Вывод пробела

return 0;

}

## 1.3 Пример работы программы

На рис. 1 продемонстрирован результат работы программы 1 упражнения.



Рисуно1. Результат программы 1 упражнения

# Упражнение 2

## 2.1. Постановка задачи

Написать и получить результат работы программы, решающей линейное уравнение ax+b=c, коэффициенты a, b, с которого задаются при выполнении.

## 2.2. Код программы

#include <stdio.h>

void main(){

double a, b, c, x; // задание переменных

puts("Enter A"); // вывод сообщения

scanf("%lf", &a); // ввод a

puts("Enter B"); // вывод сообщения

scanf("%lf", &b); // ввод b

puts("Enter C"); // вывод сообщения

scanf("%lf", &c); // ввод c

x=(c-b)/a; // вычисление x

printf("x=%lf\n",x); // вывод результата

}

## 2.3. Примеры работы программы

На рис.2, 3, 4 показаны результаты работы программы упражнения 2.

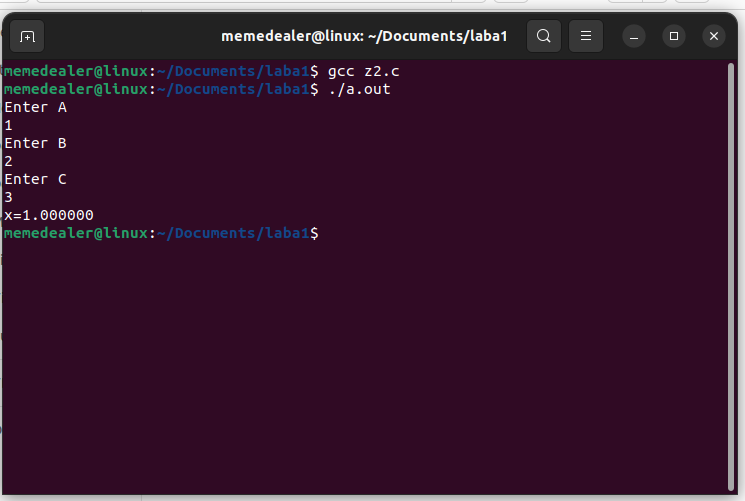


Рисунок 2. Пример 1 работы программы

Рисунок 3. Пример 2 работы программы

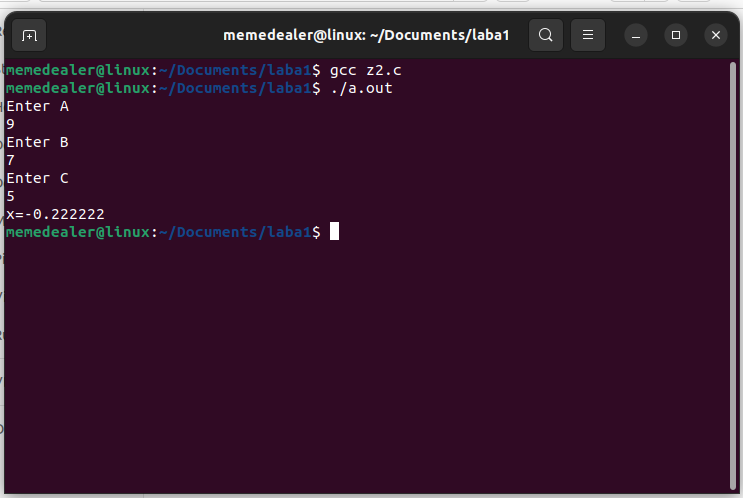
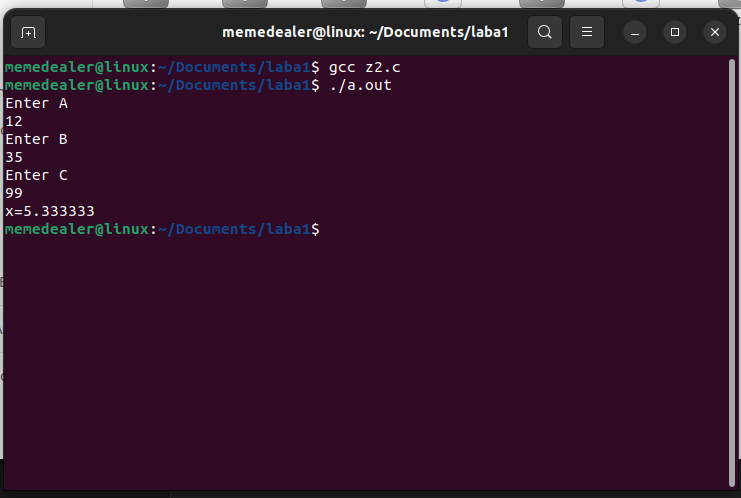


Рисунок 4. Пример 3 работы программы



# Упражнение 3

## 3.1. Постановка задачи

Вывести в терминал набор аргументов вызова исполняемого файла, переданных при вызове программы.

## 3.2 Код программы

#include <stdio.h>

int main(int argc, char \*argv[]) // argc - счетчик аргументов, argv - аргумент

{

if (argc > 1) { // проверка на существование аргументов

for (int i = 1; i < argc; i++) { // цикл перебора аргументов

printf("Argument %d: %s\n", i, argv[i]); // вывод аргументов под номером

}

}

return 0;

}

## 3.3. Пример работы программы

На рис.5 представлен результат работы программы упражнения 3.

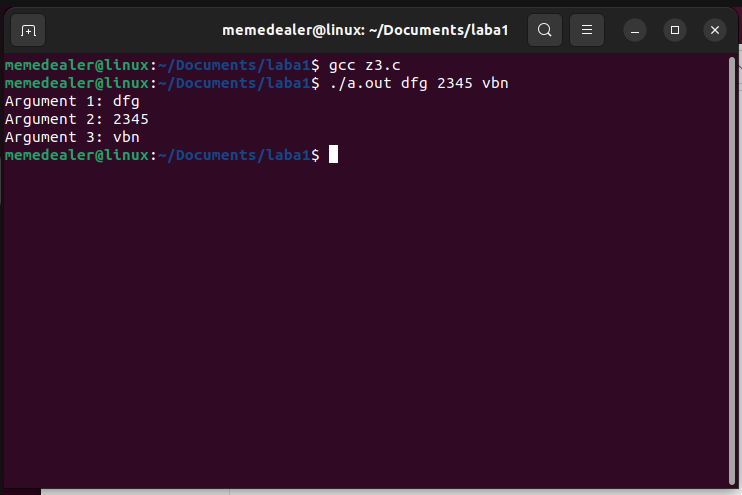


Рисунок 5. Пример работы программы упражнения 3

# Упражнение 4

## 4.1. Постановка задачи

Вывести в терминал содержимое текстового файла, название которого передается при вызове исполняемого файла.

## 4.2. Код программы

#include <stdio.h>

int main(int argc, char \*argv[]) {

if (argc == 2){ // проверка количества аргументов

FILE \*fp; // создание поинтера на файл

fp = fopen(argv[1], "r"); // открытие файла

if (fp == NULL){ // проверка файла на существование

printf("Enter a file name: ./a.out filename.txt\n"); // вывод сообщения об ошибке

}

char ch; // переменная для чтения файла

while (!feof (fp)) { // цикл для чтения файла

printf("%c", ch); // вывод символов файла

ch = fgetc(fp); // посимвольное чтение файла

}

fclose(fp); // закрытие файла

}

else{

puts("Enter a file name: ./a.out filename.txt\n"); // вывод сообщения об ошибке

}

return 0;

}

## 4.3. Пример работы программы

Содержимое текстового файла test\_programm\_4.txt (рис.6):

misha

loves

banana

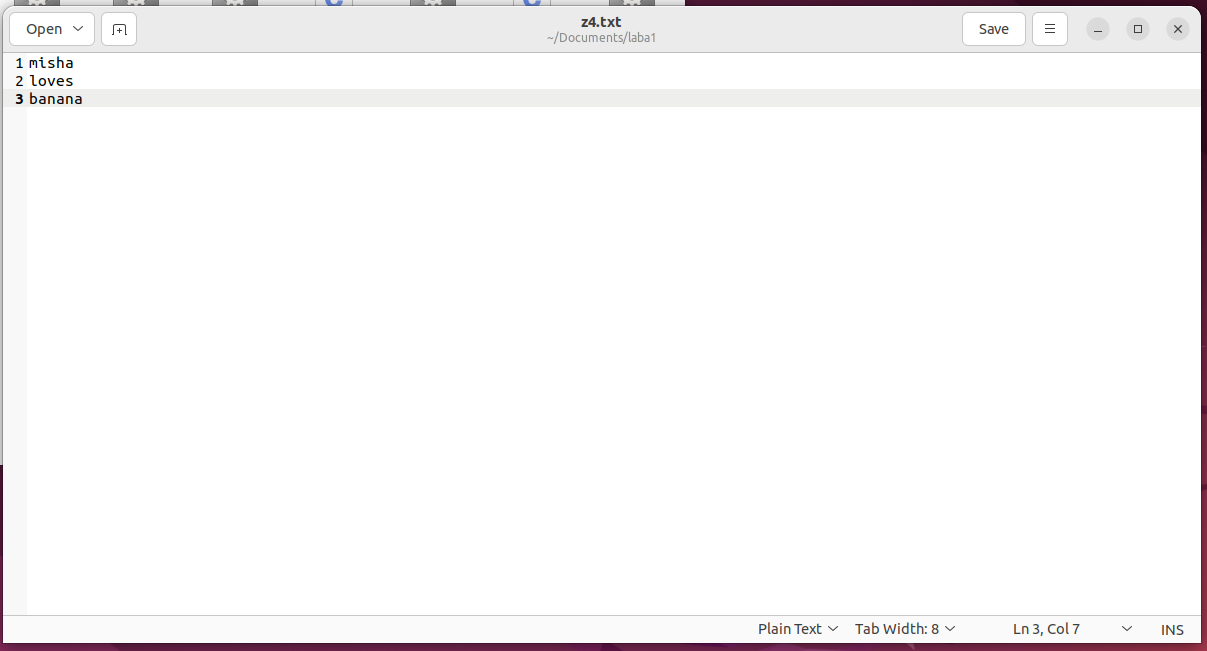


Рисунок 6. Текстовый файл к упражнению 4

На рис.7 представлен результат работы программы упражнения 4.

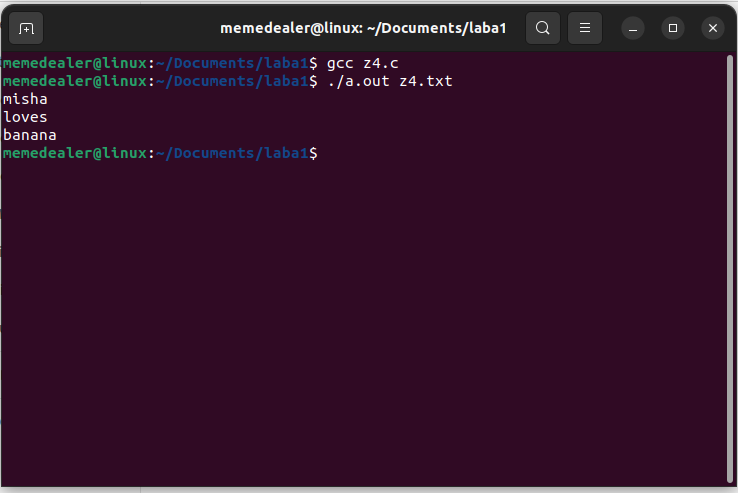


Рисунок 7. Пример работы программы упражнения 4

# Упражнение 5

## 5.1 Постановка задачи

Вывести в начало текстового файла запрошенную при работе программы информацию и после дописать содержимое файла входных данных.

## 5.2 Код программы

#include <stdio.h>

#include <string.h>

int main(int argc, char \*argv[]) {

if (argc == 2){ // проверка количества аргументов

FILE \*fp; // создание поинтера на файл

fp = fopen(argv[1], "r"); // открытие файла

if (fp == NULL){ // проверка файла на существование

printf("Enter a file name: ./a.out filename.txt\n"); // вывод сообщения об ошибке

}

int symb = 0; // переменная для подсчета количества символов

while (!feof (fp)) { // цикл для подсчета количества символов

fgetc(fp); // чтение файла посимвольно

symb++; // увеличение переменной

}

fclose(fp); // закрытие файла

char st[symb+256]; // создание массива символов

puts("Enter a sting to add to a file"); // вывод сообщения для пользователя

fgets(st, 256, stdin); // ввод текста пользователем

fp = fopen(argv[1], "r"); // открытие файла

char ch; // переменная для чтения файла

while (!feof (fp)) { // цикл для чтения файла

strncat(st, &ch, 1); // добавление символа в строку

ch = fgetc(fp); // чтение файла посимвольно

}

fclose(fp); // закрытие файла

fp = fopen(argv[1], "w"); // открытие файла и стирание информации в нем

fprintf(fp, "%s", st); // запись в файл

fclose(fp); // закрытие файла

}

else{

puts("Enter a file name: ./a.out filename.txt\n"); // вывод сообщения об ошибке

}

return 0;

}

## 5.3 Пример работы программы

Содержимое текстового файла test.txt (рис.8):

misha

loves

banana

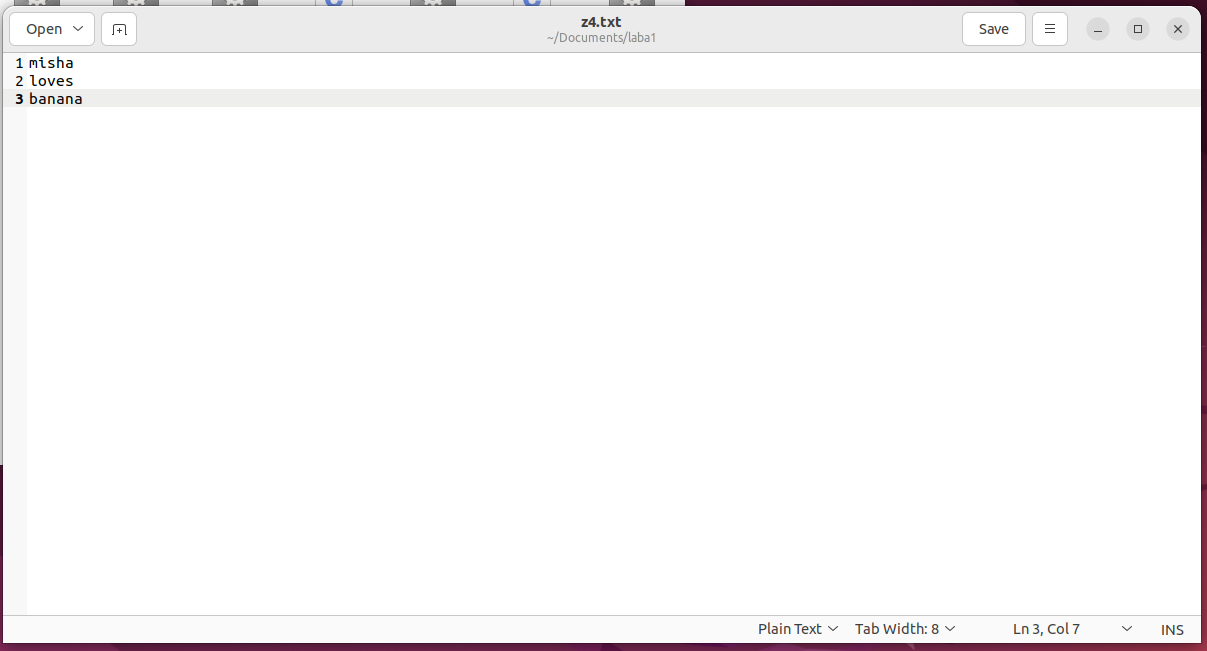


Рисунок 8. Текст к упражнению 5

На рис. 9 изображена работа программы для упражнения 5.

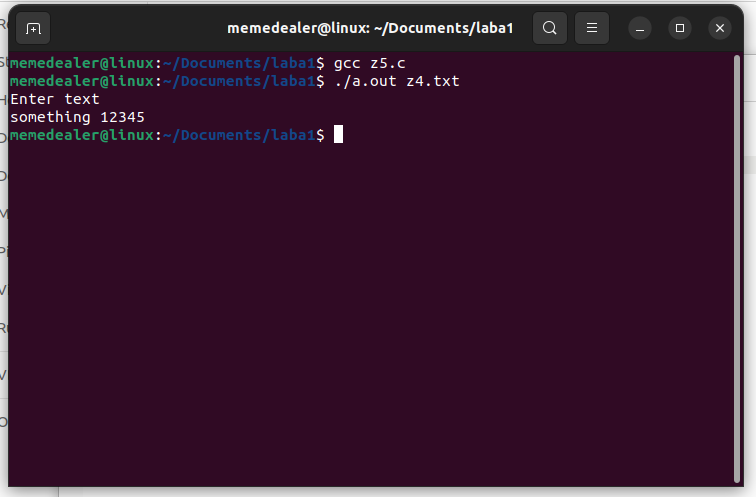


Рисунок 9. Пример работы программы упражнения 5

На рис.10 показан текстовый файл после выполнения программы.

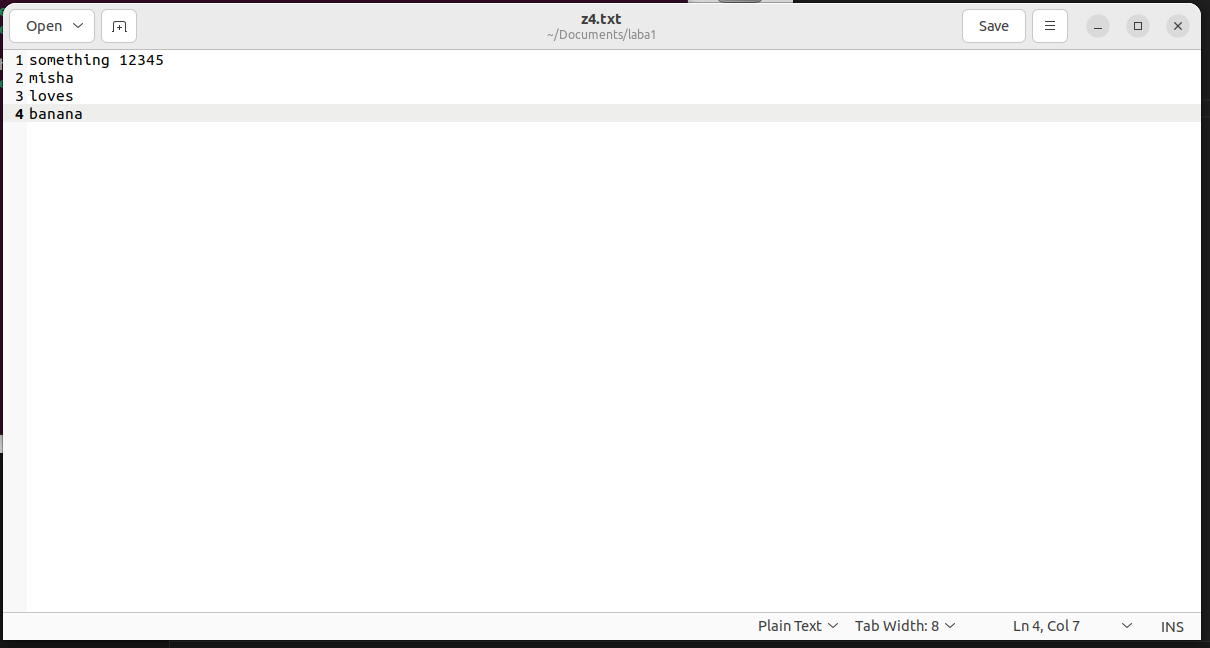


Рисунок 10. Текстовый файл после выполнения кода