Министерство образования Республики Беларусь

Учреждение образования

«БЕЛОРУССКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ ИНФОРМАТИКИ И РАДИОЭЛЕКТРОНИКИ»

Факультет компьютерных систем и сетей

Кафедра электронных вычислительных машин

Отчёт

По учебной(ознакомительной) практике

Студент:

гр.230501 Кочеров Р.С.

Руководитель:

доцент, кандидат технических наук

Одинец Д.Н

МИНСК 2023

Оглавление

[Постановка задачи 3](#_Toc139357188)

[Листинг 3](#_Toc139357189)

[Пример выполнения программы 5](#_Toc139357190)

[Заключение: 5](#_Toc139357191)

# **Постановка задачи**

***Вариант 23***. Зашифровать данный текст с помощью шифра Цезаря.

# **Листинг**

#include "Roman.h";

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

{

int n;

char buffer[1024];

printf("Enter your string: ");

input\_string(buffer, 1024);

FILE\* f = NULL;

fvvod(f, buffer, argc, argv);//ввод в файл

printf("enter how many characters to shift:");

n = vvod();//кол-во на сколько сдвинетс

encrypt(n, argc, argv);//сам шифр

fvivod(f, argc, argv);

return 0;

}

Библиотека Roman.h:

#define \_CRT\_SECURE\_NO\_WARNINGS

#include <stdlib.h>

#include <io.h>

#include <fcntl.h>

#include <sys/types.h>

#include <sys/stat.h>

#include <stdio.h>

#include <share.h>

#define ENG 26

char\* input\_string(char\* buffer, int buffer\_size) {

if (fgets(buffer, buffer\_size, stdin) != NULL) {

int length = strlen(buffer);

if (buffer[length - 1] == '\n') {

buffer[length - 1] = '\0';

}

}

return buffer;

}

void encrypt(int n, int argc, char\*\* argv[])

{

FILE\* fp1, \* fp2;

fopen\_s(&fp1, argv[1], "r");

fopen\_s(&fp2, "output.txt", "w");

int flag;

char c;

c = getc(fp1);

while (!feof(fp1))

{

flag = 0; //обработан ли текущий символ

if (c >= 'A' && c <= 'Z')

{

c = c + (n % ENG);

if (c > 'Z') c = 'A' + (c - 'Z') - 1;

fprintf(fp2, "%c", c);

flag = 1;

}

if (c >= 'a' && c <= 'z')

{

c = c + (n % ENG);

if (c > 'z') c = 'a' + (c - 'z') - 1;

fprintf(fp2, "%c", c);

flag = 1;

}

if (!flag) fprintf(fp2, "%c", c);

c = getc(fp1);

}

fclose(fp1);

fclose(fp2);

}

int vvod()

{

int var;

while ((scanf\_s("%d", &var) == 0) || getchar() != '\n')

{

printf("wrong input! try again\n");

rewind(stdin);

}

return var;

}

int fvvod(FILE\* f,char\* arr, int argc, char\*\* argv[])

{

if ((f = fopen(argv[1], "w")) == NULL)

{

printf("Cannot open file.\n");

return 0;

}

fputs(arr, f);

fclose(f);

return 0;

}

int fvivod(FILE\* f, int argc, char\*\* argv[])

{

char c[1000];

if ((f = fopen("output.txt", "r")) == NULL)

{

printf("Cannot open file.\n");

return 0;

}

while (!feof(f))

{

if (fgets(c, 1000, f) != NULL)

printf("%s", c);

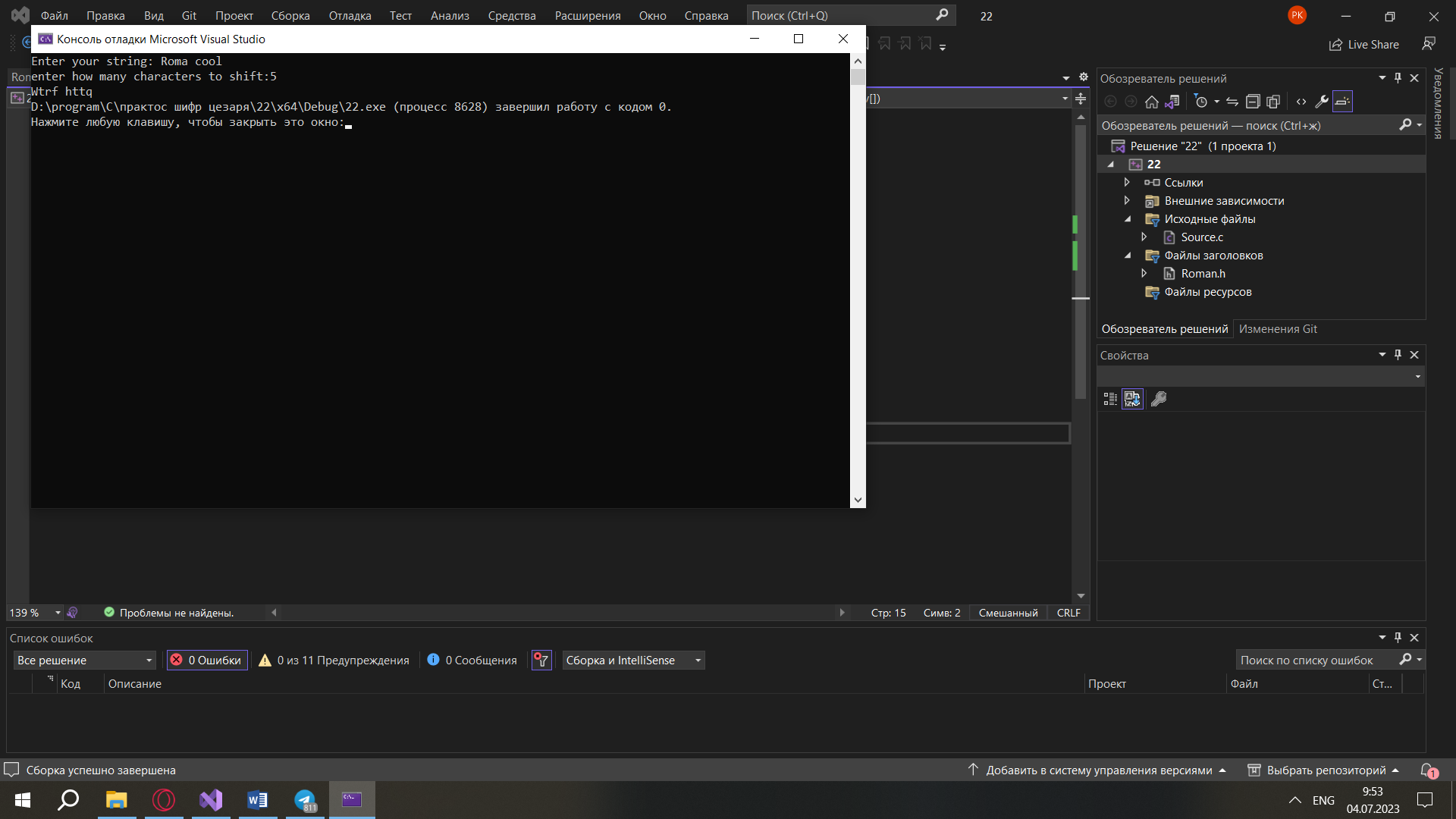
}

fclose(f);

return 0;

}

# **Пример выполнения программы**



# **Заключение:** Опишу по порядку, что происходит в функции encrypt. Открывает входной файл «input.txt» для чтения, открываем (или если он отсутствует, то создаем)  выходной файл «output.txt» для записи. Функцией getc() будем по одному считывать символы из входного файла. Запускаем цикл с предусловием while (!feof(fp1)), функция в скобках проверяет, не достигли ли мы конца файла. В if’ах проверяем принадлежность считанного символа одной из групп символов, если считанный символ ‘c’ — это английская буква, то выполняем ее шифрование, сдвигаем: c = c + (n % ENG). Остаток от деления на количество букв в алфавите берем для того, чтобы при n >= ENG убрать лишний «круг(и)» прохода по алфавиту. Если зашифрованная буква вышла за границы алфавита, то делаем круг и возвращаемся к началу: if (c > ‘Z’) c = ‘A’ + (c — ‘Z’) — 1;. Записываем символ в выходной файл. Существенных ограничений нету. В дальнейшем можно добавить несколько языков по тому же алгоритму.