Правительство Российской Федерации

Федеральное государственное автономное образовательное учреждение
высшего образования «Национальный исследовательский университет
«Высшая школа экономики»

Кафедра «Компьютерная безопасность»

ОТЧЕТ
К ЛАБОРАТОРНОЙ РАБОТЕ №2

по дисциплине

«Языки программирования»

Работу выполнила
студентка группы
СКБ-232 _____ Д.В. Иванова
подпись, дата

Работу проверил _____ С.А. Булгаков
подпись, дата

Москва 2023

Содержание

1 Постановка задачи

Необходимо было улучшить программу из Лабораторной работы №1 следующим образом: если при запуске программы ей передается параметр командной строки ‘k‘ содержащий целочисленный аргумент, то его значение используется в качестве сдвига в алгоритме.

Поскольку остальные улучшения программы были оставлены на усмотрение исполнителя, то я сделала следующие изменения:

1. Выделила часть кода, отвечающую за шифрование шифром Цезаря в отдельную функцию Caesar, типа int, получающую на вход символ ch типа char и число shift типа int и возвращающую зашифрованный символ ch.

2. Написала функцию is_number типа int, получающую на вход указатель на символ char* ch, возвращающую значение 1, если строка, начинающаяся в переданном адресе, является целым числом, иначе возвращающую 0. Функция была написана для проверки параметра ‘k‘.

2 Описание функции main

2.1 Ввод и работа с аргументами

- Если пользователь не вводил имя файла как аргумент при запуске программы, то после запуска программу пользователю предлагается ввести сообщение, которое он хочет закодировать и сдвиг, на который сдвинется алфавит вперёд.
- Если же пользователь ввёл название файла и такой файл существует, то шифруемый текст считывается из файла, сдвиг пользователь вводит в ответ на запрос программы, после чего создаётся файл enc.txt в который записывается зашифрованное сообщение.
- Если пользователь ввёл только сдвиг в качестве аргумента, то шифруемое сообщение пользователю предложат ввести после слов ‘Input message:‘.
- Если пользователь в качестве аргументов ввёл и имя файла и число, на которое должен быть совершён сдвиг, при том не важно в каком порядке, то программа создаст файл enc.txt и запишет в него зашифрованное сообщение.

2.2 Описание функции Caesar

Функция Caesar реализует агоритм шифрования Цезаря определённого элемента. Алгоритм шифрования Цезаря проверяет вводимый символ на то, заглавная это буква или строчная. После определения, ищется место этой буквы в алфавите, прибавляется сдвиг и проверяется, что не случилось выхода из алфавита, после чего в алфавите ищется буква с новым номером и запоминается как буква зашифрованного слова. Таким образом, шифруется каждая буква вводимого сообщения и получается зашифрованное сообщение.

2.3 Описание функции is_number

Функция получает на вход ссылку на элемент char* str. Дальше в цикле, пока не встречен символ перехода на следующую строчку или символ окончания строки, проверяется является ли данный символ цифрой. Если проверка не пройдена - функция возвращает 0, иначе, если за время работы цикла программа не выведет 0, то будет выведена 1.

3 Тестирование

3.1 Проверка работы с файлом

Запишем в файл mes.txt "Hello проверим, что файла enc.txt изначально не было. Запустим программу с аргументом mes.txt, введём сдвиг 5. Появился файл enc.txt с зашифрованным по алгоритму шифра Цезаря со сдвигом 5 словом "Hello".

```
$ cat mes.txt
Hello
$ ls
a.out  Caesar.c  mes.txt  README.md
$ ./a.out mes.txt
Input shift:5
$ cat enc.txt
Mjqqt
```

3.2 Проверка работы с пользовтельским вводом

После запуска программы пользователся просят ввести сдвиг и сообщение, после чего выводится "Encrypted message:"и сообщение, зашифрованное сообщение с ведённым сдвигом по алгоритму шифрования Цезаря.

```
$ ./a.out
Input shift:5
Input message:Hello
Encrypted message:Mjqqt
```

3.3 Проверка работы с целым числом без файла

Если пользователь вводит в качестве параметра целое число, то программа просит его ввести сообщение для шифрования и выводит зашифрованное, со сдвигом на подаваемое число, сообщение.

```
$ ./a.out  5
Input message:Hello
Encrypted message:Mjqqt
```

3.4 Проверка работы с целым числом и файлом

Если пользователь вводит в качестве параметров целое число и файл, то программа создаёт файл enc.txt, если такового не было, и записывает в него зашифрованное сообщение.

```
$ ls
a.out  CoolCaesar.c  msg.txt  README.md
$ ./a.out  msg.txt 5
$ ls
a.out  CoolCaesar.c  enc.txt  msg.txt  README.md
$cat enc.txt
Mjqqt
```

3.5 Проверка вывода сообщения об ошибке открытия файла

Если файла, который пользователь указал в аргументе к программе нет, то программа выводит ошибку с соответсвующим предупреждением.

```
$ ./a.out what.txt
no such file
fopen(): No such file or directory
```

3.6 Проверка вывода ошибки о нецелом числе или отсутствии файла для чтения

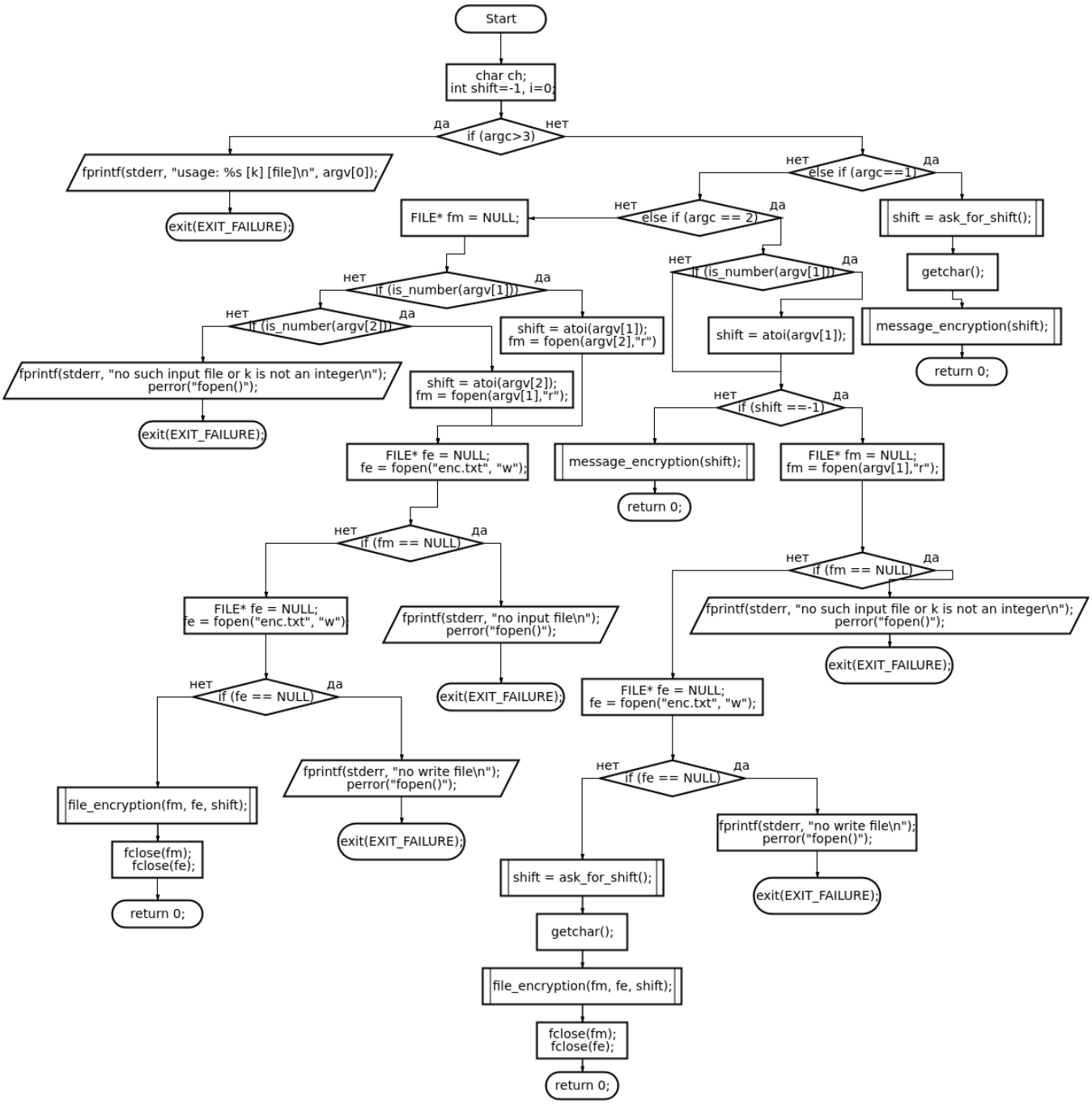
Если пользователь вводит в качестве параметра нецелое число, то программа должна завершить работу с ошибкой.

```
$ ./a.out  msg.txt 5.5
no such input file or k is not an integer
fopen(): Success
```

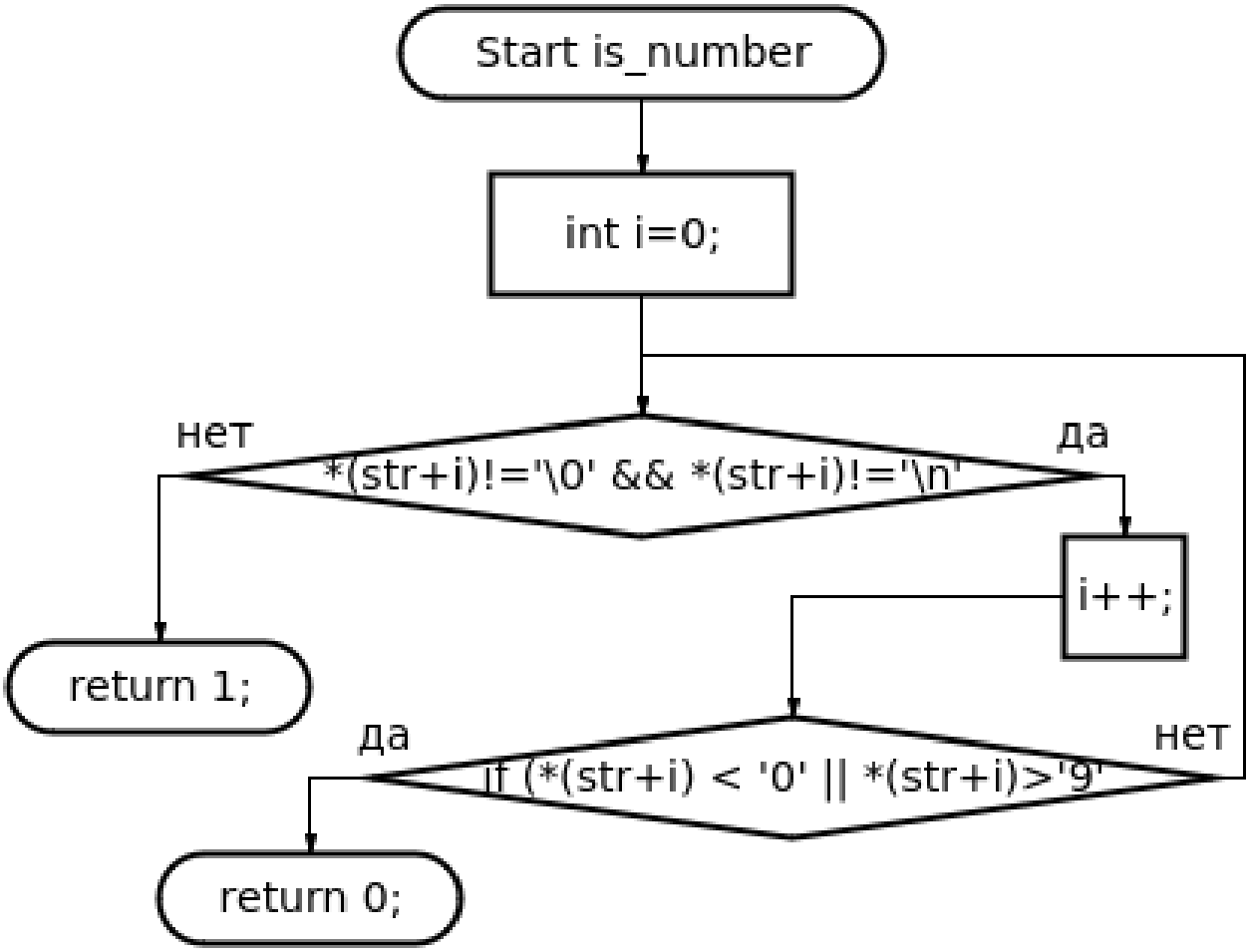
```
$ ./a.out  5.5
no such input file or k is not an integer
fopen(): No such file or directory
```

Приложение А

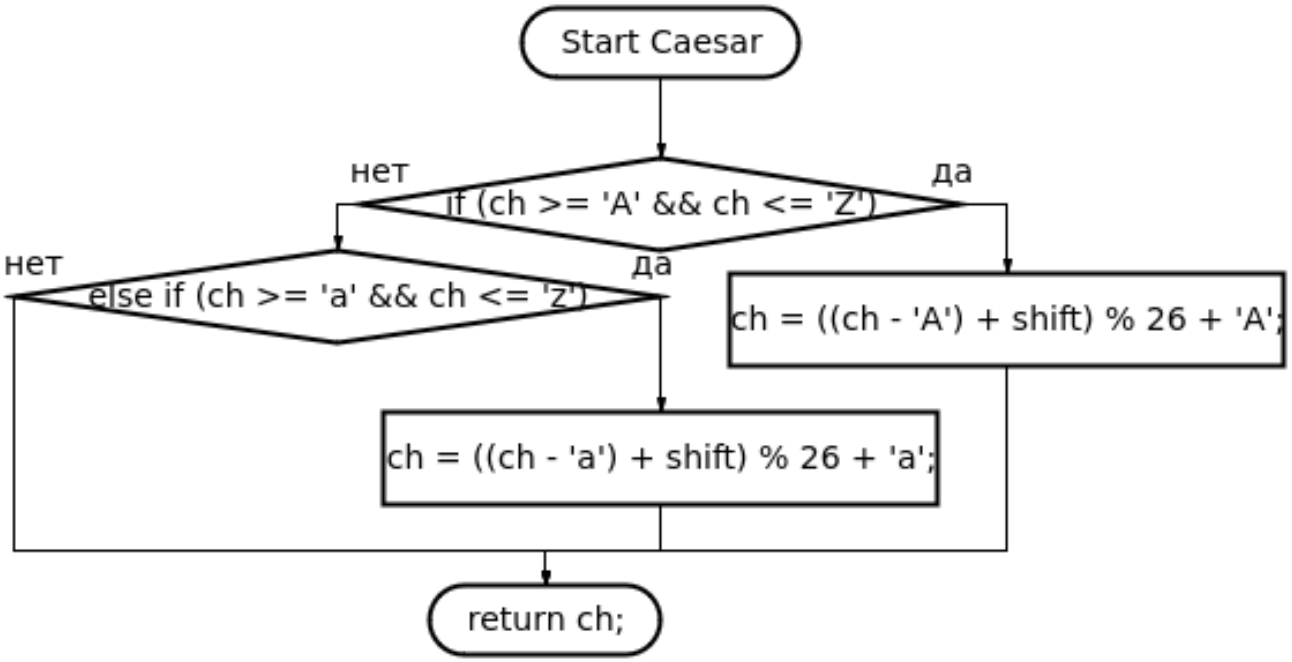
Блок-схема *main*



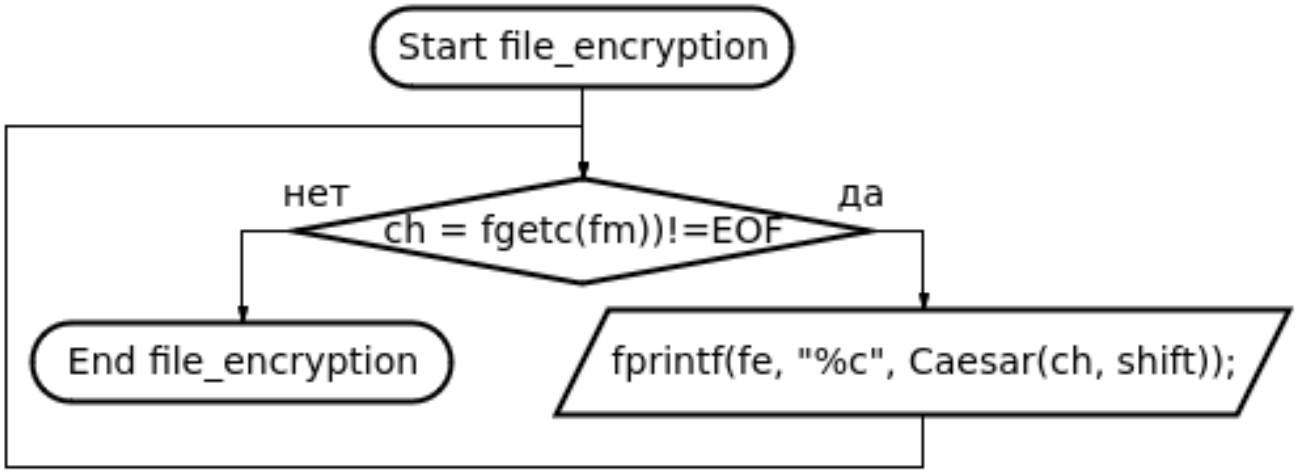
Блок-схема *is_ number*



Блок-схема *Caesar*



Блок-схема *file_ encryption*



Приложение Б

Код программы

```
#include <stdio.h>
#include <stdlib.h>
#include <errno.h>

char ch;
int shift=-1, i=0;

char Caesar(char ch, int shift){
    if (ch >= 'A' && ch <= 'Z')
        ch = ((ch - 'A') + shift) % 26 + 'A';
    else if (ch >= 'a' && ch <= 'z')
        ch = ((ch - 'a') + shift) % 26 + 'a';
    return ch;
}

int ask_for_shift(){
    printf("Input shift:");
scanf("%d", &shift);
return shift;
}

void message_encryption(int shift){
    printf("Input message:");
while ((ch = getchar()) != '\n' && ch != EOF){
if (i==0){
    printf("Encrypted message:");
i=1;
    }
printf("%c", Caesar(ch, shift));
}
printf("\n");
}

void file_encryption(FILE* fm, FILE* fe, int shift){
    while ((ch = fgetc(fm))!=EOF){
fprintf(fe, "%c", Caesar(ch, shift));
}
}

int is_number(char* str){
    for (int i=0; *(str+i)!='\0' && *(str+i)!='\n'; i++){
        if (*(str+i) < '0' || *(str+i)>'9')
            return 0;
    }
    return 1;
}

int main(int argc, char **argv){

if (argc>3){
fprintf(stderr, "usage: %s [k] [file]\n", argv[0]);
exit(EXIT_FAILURE);
}
else if (argc == 1){
shift = ask_for_shift();
getchar();
message_encryption(shift);
}
else if (argc == 2){
if (is_number(argv[1]))
    shift = atoi(argv[1]);
if (shift ==-1){
    FILE* fm = NULL;
    fm = fopen(argv[1],"r");
    if (fm == NULL){
    fprintf(stderr, "no such input file or k is not an integer\n");
    perror("fopen()");
    exit(EXIT_FAILURE);
    }
    FILE* fe = NULL;
    fe = fopen("enc.txt", "w");
    if (fe == NULL){
                    fprintf(stderr, "no write file\n");
                    perror("fopen()");
                    exit(EXIT_FAILURE);
```

```
                }
    shift = ask_for_shift();
    getchar();
    file_encryption(fm, fe, shift);
    fclose(fm);
    fclose(fe);
}
else
    message_encryption(shift);
}
else {
    FILE* fm = NULL;
if (is_number(argv[1])){
    shift = atoi(argv[1]);
    fm = fopen(argv[2],"r");
}
else if (is_number(argv[2])){
    shift = atoi(argv[2]);
    fm = fopen(argv[1],"r");
}
else {
    fprintf(stderr, "no such input file or k is not an integer\n");
perror("fopen()");
exit(EXIT_FAILURE);
}
FILE* fe = NULL;
fe = fopen("enc.txt", "w");
if (fm == NULL){
fprintf(stderr, "no input file\n");
perror("fopen()");
exit(EXIT_FAILURE);
}
if (fe == NULL){
                fprintf(stderr, "no write file\n");
                perror("fopen()");
                exit(EXIT_FAILURE);
            }
file_encryption(fm, fe, shift);
fclose(fm);
fclose(fe);
}
return 0;
}
```