Федеральное государственное автономное
образовательное учреждение
высшего образования
«СИБИРСКИЙ ФЕДЕРАЛЬНЫЙ УНИВЕРСИТЕТ»

Институт космических и информационных технологий
институт

Кафедра «Информатика»
кафедра

ОТЧЕТ О ПРАКТИЧЕСКОЙ РАБОТЕ № 3

Работа со строками
Тема

Руководитель _____ Черниговский А.С.
Подпись, дата Инициалы, Фамилия

Студент КИ19-17/1Б, №031939174 _____ А.К.Никитин
Номер группы, зачетной книжки Подпись, дата Инициалы, Фамилия

Красноярск 2020

1 Цель

Ознакомиться с базовым синтаксисом языка СИ, его стандартом кодирования, принципом работы указателей и динамическими массивами и написать программу по варианту.

2 Задачи

Для выполнения лабораторной работы необходимо выполнить следующие задачи:

1) написать программу в соответствии с заданием;

2) отформатировать исходный код программы согласно стандарту оформления исходного кода;

3) выполнить требования предыдущих пунктов;

4) для хранения строк использовать динамические массивы символов

5) (размер массива определяется в процессе ввода);

6) организовать повтор программы по желанию пользователя;

7) добавить проверку входных аргументов на корректность;

8) выполнить требования предыдущих пунктов;

9) реализовать меню пользователя, состоящее как минимум из 4-х пунктов

10) (ввод данных, обработка данных, вывод результата на экран, выход);

11) для корректной работы меню организовать промежуточное хранение результата

12) разбить программу на функции;

13) организовать чтение данных и запись результата в файл формата txt.

3 Описание задания

Дан текст, который может содержать буквы английского алфавита. Напишите программу, предназначенную для шифрования и расшифровки текста, используя шифр Вернама. Перед шифрованием удалите из текста все знаки препинания и повторяющиеся пробелы, приведите символы к нижнему регистру.

4 Ход выполнения

Ниже представлен листинг программы по заданию.

Листинг 1 – Шифрование и дешифровка шифром Вернама

```
#include <stdio.h>
#include <stdlib.h>
#include <locale.h>
#include <string.h>

#define NUMBERS_IN_ALPHABET 53
#define MAX_FILE_STRING_NUMBER 1000
#define CHAR_A 65
#define CHAR_a 97
#define CHAR_Z 90

// Ввод только натурального числа
void inputNat(int* number)
{
    while (!scanf("%d", number))
    {
        fflush(stdin);
        printf("Введите корректные данные!\n");
    }
    if (*number < 0)
    {
        printf("Число должно быть натуральным или 0!\n");
        inputNat(number);
    }
}

// Функция оставляет в строке только символы английского алфавита и пробел
void correctString(char* string)
{
    char* correctString; // Строка, вбирающая в себя все правильные символы
    int count = 0;
    int checker = 0;

    correctString = (char*) malloc(strlen(string) * sizeof(char));
    for (int i = 0; i < strlen(string); i++)
            // a-z, A-Z, ' '
```

Продолжение листинга

```
        if ((string[i] >= 65 && string[i] <= 90) || (string[i] >= 97 &&
string[i] <= 122) || (string[i] == 32))
        {
            correctString[count] = string[i];
            count++;
            checker = 1;
        }
    if (checker)
    {
        strcpy(string, correctString);
        free(correctString);
    }
    else
    {
        printf("Ваше сообщение целиком состоит из неанглийских символов!\n");
    }
}

// Ввод пользователем строки произвольной длины и ее корректировка предыдущей
функцией
void inputCorrectString(char** word)
{
    int count = 0;
    char inputChar;

    fflush(stdin);

    while(1)
    {
        inputChar = getchar();
        if (inputChar == '\n')
            break;
        else
        {
            *word = realloc(*word, count + 1);
            (*word)[count] = inputChar;
            count++;
        }
    }
    (*word)[count] = '\0';
```

Продолжение листинга

```
    correctString(*word);
}

// Создание массива всех символов алфавита
int* alphabet()
{
    int* alphabetNum;
    int count = 0;
    int ansiChar = CHAR_A;

    alphabetNum = (int*) malloc(NUMBERS_IN_ALPHABET * sizeof(int));
    for (; count < NUMBERS_IN_ALPHABET - 1; count++, ansiChar++)
    {
            // Переход из символа Z к символу a
        if (ansiChar == CHAR_Z + 1)
            ansiChar = CHAR_a;

        alphabetNum[count] = ansiChar;

    }
    alphabetNum[count] = ' ';
    return alphabetNum;
}

// "Подгон" длины ключа к длине сообщения. Пример: Hello world
//                                                  keykeykeyke
char* keyLengthen(char* key, int length)
{
    int lenKey = strlen(key);
    int count = 0;
    char* newKey;

    newKey = (char*) malloc(length * sizeof(char));
    // Ключ меньше текста
    if (strlen(key) < length)
    {
        for (int i = 0; i < length; i++, count++)
        {
            // Обнуление счетчика для повторного запуска ключа по кругу
            if (count == lenKey)
```

Продолжение листинга

```
                count = 0;

            newKey[i] = key[count];
        }
    }
    // Ключ больше либо равен тексту
    else if (strlen(key) >= length)
    {
        strncpy(newKey, key, length);
        return newKey;
    }
    return newKey;
}

// Ставит соответствие между номером символа из ANSI и ее порядковым номером в
алфавите. Пример: 'b'(98 символ) = 1
int ansi2alphabet(int ansiCode)
{
    int* allCharsNumbers;
    int alphabetCode;
    allCharsNumbers = alphabet();

    for (int i = 0; i < NUMBERS_IN_ALPHABET; i++)
        if (allCharsNumbers[i] == ansiCode)
        {
            alphabetCode = i;
            break;
        }
    return alphabetCode; // Возвращает номер символа в алфавите
}

// Кодирование сообщений шифром Вернама
char* encode(char* text, char* key)
{
    int* allCharsNumbers;
    int length;
    char* longKey;
    int tempCharNumber;
    int* encryptedTextNum = NULL;
    char* encryptedTextStr = NULL;
```

Продолжение листинга

```
    allCharsNumbers = alphabet();
    length = strlen(text);
    longKey = keyLengthen(key, length);

    encryptedTextNum = (int*) malloc(length * sizeof(int));
    encryptedTextStr = (char*) malloc(length * sizeof(char));

    for (int i = 0; i < length; i++)
    {
        // Остаток от деления суммы кодов двух символов
        encryptedTextNum[i] = (ansi2alphabet((int) text[i]) +
ansi2alphabet((int) longKey[i])) % NUMBERS_IN_ALPHABET;

        // Преобразование остатка в символ
        encryptedTextStr[i] = (char) allCharsNumbers[encryptedTextNum[i]];
    }
      free(encryptedTextNum);
    return encryptedTextStr;
}

// Декодирование сообщений шифром Виженера
char* decode(char* text, char* key)
{
    int* allCharsNumbers;
    int length;
    char* longKey;
    int tempCharNumber;
    int* decryptedTextNum = NULL;
    char* decryptedTextStr = NULL;

    allCharsNumbers = alphabet();
    length = strlen(text);
    longKey = keyLengthen(key, length);

    decryptedTextNum = (int*) malloc(length * sizeof(int));
    decryptedTextStr = (char*) malloc(length * sizeof(char));

    for (int i = 0; i < length; i++)
    {
            // Формула для дешифровки
```

Продолжение листинга

```
        decryptedTextNum[i] = (ansi2alphabet((int) text[i]) +
NUMBERS_IN_ALPHABET -
                               ansi2alphabet((int) longKey[i])) %
NUMBERS_IN_ALPHABET;

        decryptedTextStr[i] = (char) allCharsNumbers[decryptedTextNum[i]];
    }
      free(decryptedTextNum);
    return decryptedTextStr;
}

int main()
{
    char* message = NULL;
    char* key = NULL;
    int userChoice;
    int fileChoice;

    setlocale(LC_ALL, "");

    enum Case {textInput = 1, keyInput, encrypt, decrypt, exit_prog};

    do
    {
        printf("Введите ваш выбор:\n"
               "1. Ввести текст.\n"
               "2. Ввести ключ.\n"
               "3. Произвести шифровку\n"
               "4. Произвести дешифровку\n"
               "5. Выход\n");
        inputNat(&userChoice);

        if (userChoice < 1 || userChoice > 5)
        {
            printf("Введите значение от 1 до 5!\n");
            continue;
        }

        switch (userChoice)
        {
            // Ввод сообщения
```

Продолжение листинга

```
            case (textInput):
            {
                printf("Хотите ли вы считать сообщение с консоли или с
файла?(1/0)\n");
                inputNat(&fileChoice);

                if (fileChoice == 1)
                {
                    printf("Введите текст для шифрования:\n");
                    inputCorrectString(&message);
                }
                else if (fileChoice == 0)
                {
                            // Работа с текстовым файлом
                    FILE *fileMessage;

                    message = (char*) malloc(MAX_FILE_STRING_NUMBER *
sizeof(char));

                    fileMessage =
fopen("C:\\Users\\alekc\\Desktop\\Learning\\OP2\\Laba3\\message.txt", "r");
                    fgets(message, MAX_FILE_STRING_NUMBER, fileMessage);
                    fclose(fileMessage);

                    correctString(message);
                }
                else
                    printf("Вы должны были ввести 1 или 0(\n");
                break;
            }
              // Ввод ключа
            case (keyInput):
            {
                printf("Хотите ли вы считать сообщение с консоли или с
файла?(1/0)\n");
                inputNat(&fileChoice);

                if (fileChoice == 1)
                {
                    printf("Введите ключ:\n");
                    inputCorrectString(&key);
```

Продолжение листинга

```
                }
                else if (fileChoice == 0)
                {
                    FILE *fileMessage;
                    key = (char*) malloc(MAX_FILE_STRING_NUMBER * sizeof(char));

                    fileMessage =
fopen("C:\\Users\\alekc\\Desktop\\Learning\\OP2\\Laba3\\key.txt", "r");
                    fgets(key, MAX_FILE_STRING_NUMBER, fileMessage);
                    fclose(fileMessage);

                    correctString(key);
                }
                else
                    printf("Вы должны были ввести 1 или 0(\n");
                break;
            }
              // Шифрование
            case (encrypt):
            {
                char* encodeMessage;

                if (message == NULL || key == NULL)
                {
                    printf("Сначала введите сообщение и ключ!\n");
                    continue;
                }

                encodeMessage = encode(message, key);
                puts(encodeMessage);
                break;
            }
              // Дешифровка
            case (decrypt):
            {
                char* decodeMessage;

                if (strlen(message) == 0 || strlen(key) == 0)
                {
                    printf("Сначала введите сообщение и ключ!\n");
                    continue;
```

Продолжение листинга

```
                }

                decodeMessage = decode(message, key);
                puts(decodeMessage);
                break;
            }
                  // Выход
            case (exit_prog):
            {
                free(message);
                free(key);
                exit(1);
                break;
            }

        }

    }while (1);
}
```

5 Результат

Ниже представлены скриншоты с консольным выводом.

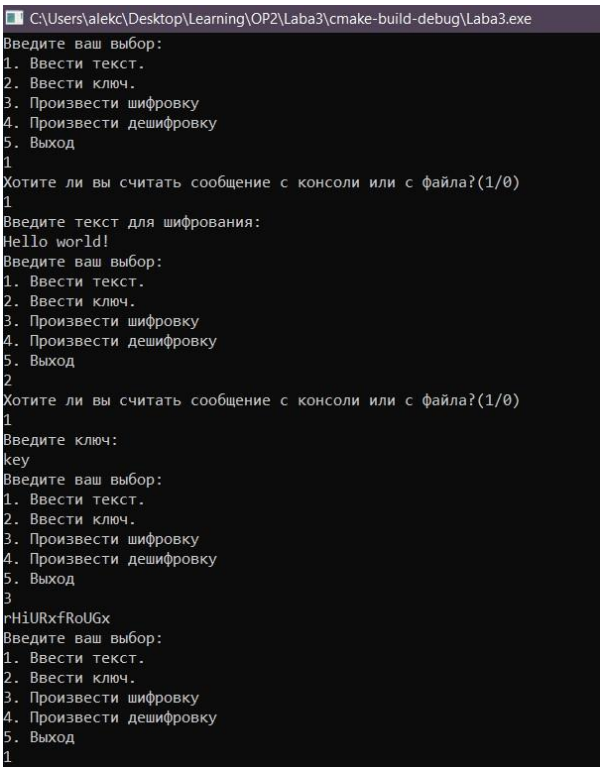

Рисунок 1 – Шифрование сообщения

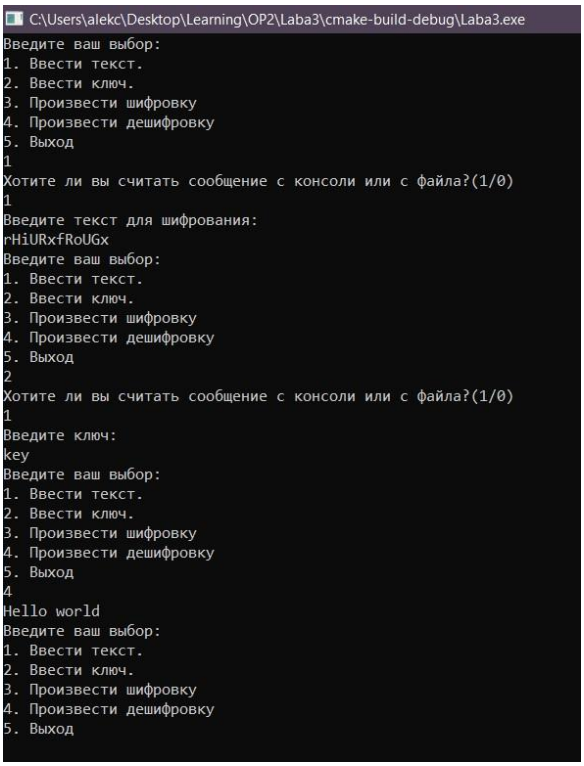

Рисунок 2 – Дешифровка сообщения

6 Выводы

По окончании работы были выполнены следующие задачи:

1) ознакомление с алгоритмами шифрования и принципами их работы;

2) изучение возможности работы с текстовыми файлами в языке Си;

3) приобретены основные знания работы со строками;