

Отчет по лабораторной работе № 11 по курсу “Фундаментальная информатика”

Студент группы М80-109Б-22 Потехин Фёдор Максимович, № по списку 15

Контакты @potehafm

Работа выполнена: «25» декабря 2022г.

Преподаватель: каф. 806 Сысоев Максим Алексеевич

Отчет сдан « » _____ 20__ г., итоговая оценка _____

Подпись преподавателя _____

1. Тема: Системы программирования на Си
2. Цель работы: Научиться основам программирования на Си
3. Задание: Раскодировать текста, закодированный по Цезарю с переменным ключом, равным номеру слова в строке.
4. Оборудование (студента):
Процессор *Intel Core i3-7100U @ 4x 2.4GH* с ОП 12288 Мб, НМД 512 Гб. Монитор 1920x1080
5. Программное обеспечение (студента):
Операционная система семейства: *linux*, наименование: *ubuntu*, версия 20.04
интерпретатор команд: *bash* версия 4.4.19.
Система программирования -- версия --, редактор текстов *emacs* версия 25.2.2
6. Идея, метод, алгоритм:

Имея входные данные опеределять строку и делать сдвиг по алфавиту в зависимости от номера строки.
7. Сценарий выполнения работы:
 - 1) Придумать реализацию поставленной задачи
 - 2) Воплотить идею на языке Си

8. Распечатка протокола:

```
#include <stdio.h>
#include <stdbool.h>
#include <assert.h>
#include <stdlib.h>

typedef enum State {
    NewWord,
    Rubbish,
    Word
} State;

bool is_NewWord (char symbol) {
    return (symbol == '\n' || symbol == '\t' || symbol == ',' || symbol == ' ') ? true : false;
}

bool is_Letter (char symbol) {
    return (symbol >= 'a' && symbol <= 'z') ? true : false;
}

void printWord (int size, char word [size]) {
    for (int i = 0; i < size; i++) {
        printf("%c", word [i]);
    }
    printf("\n");
}

void CaesarDecoder (int size, char word [size], int right_move) {
    if (right_move == 26) {
        printWord(size, word);
        return;
    }
    char alphabet [26] = {'a', 'b', 'c', 'd', 'e', 'f', 'g', 'h', 'i', 'j', 'k', 'l', 'm',
        'n', 'o', 'p', 'q', 'r', 's', 't', 'u', 'v', 'w', 'x', 'y', 'z'};
    int letter_index;
    for (int i = 0; i < size; i++) {
        letter_index = (((word [i] - 'a') + right_move) > 25) ?
            ((word [i] - 'a') + right_move) % 26 : ((word [i] - 'a') + right_move);
        word [i] = alphabet [letter_index];
    }
    printWord(size, word);
}

int main () {
    State state = NewWord;
    char symbol = '0', *word;
    int length = 0, max_length = 100, word_index = 1;
    word = (char *) malloc(max_length);
    assert(is_NewWord(' '));
    assert(!is_NewWord('a'));
    assert(is_Letter('a'));
    assert(!is_Letter('%'));
    while ((symbol = getchar()) != EOF) {
        switch (state) {
            case NewWord:
                if (is_NewWord(symbol)) {
                    length = 0;

```

```

        break;
    }
    if (is_Letter(symbol)) {
        word[length] = symbol;
        length++;
        state = Word;
    }
    else {
        state = Rubbish;
    }
    break;
case Rubbish:
    if (is_NewWord(symbol)) {
        state = NewWord;
    }
    break;
case Word:
    if (is_Letter(symbol)) {
        if (length >= max_length) {
            max_length *= 2;
            word = (char *) realloc(word, max_length);
        }
        word[length] = symbol;
        length++;
        break;
    }
    if (is_NewWord(symbol)) {
        if (word_index > 26) {
            word_index = 1;
        }
        CaesarDecoder(length, word, word_index);
        word_index++;
        length = 0;
        state = NewWord;
        break;
    }
    state = Rubbish;
    length = 0;
    break;
default:
    break;
}
}
printf("\n");
if (length != 0) {
    CaesarDecoder(length, word, word_index);
}
free(word);
return 0;
}

```

9. Дневник отладки

№	Лаб. или дом.	Дата	Время	Событие	Действие по исправлению	Примечание

10. Замечания автора по существу работы
Замечаний нет

11. Выводы. Можно реализовать решения проблем, связанных с криптографией, которые могут значительно улучшить процесс

Подпись студента _____