

Федеральное агентство связи
Федеральное государственное бюджетное образовательное учреждение
высшего образования
«Сибирский государственный университет телекоммуникаций и
информатики»
(СибГУТИ)

Расчетно-графическое задание по дисциплине
«Программирование»

Выполнил: студент гр. ИА-331

Сергеев Павел Алексеевич

Проверил:

Лебеденко Людмила Фёдоровна

ОЦЕНКА, подпись

Новосибирск 2024

Требования:

1. Чтение и шифрование текстовых данных:

- Программа должна читать текстовые данные из первого файла.
- Программа должна использовать алгоритм XOR (исключающее ИЛИ) для шифрования текста.

2. Вывод зашифрованного текста и проверка:

- Вывод зашифрованного текста на экран.
- Обратное шифрование (дешифровка) текста.
- Проверка и вывод результата, совпадает ли текст до шифрования с текстом после дешифровки.

3. Работа с файлами:

- Программа должна открывать и читать данные из двух файлов: первый файл с исходным текстом, второй — с зашифрованным текстом (если такой файл имеется).

4. Криптостойкость шифра.

5. Динамическое выделение памяти:

- Использование динамического выделения памяти для считывания текста.

6. Использование динамической библиотеки:

- Функции для шифрования и дешифровки вынесены в динамическую библиотеку.

Методы и алгоритмы:

1. Алгоритм решения

- Разложить номер билета на цифры.
- Вычислить сумму цифр первой половины билета.
- Вычислить сумму цифр второй половины билета.
- Сравнить полученные суммы.

2. Работа с файлами:

- ``fopen`` для открытия файлов.
- ``fread`` или ``fgets`` для чтения данных.
- ``fclose`` для закрытия файлов.

3. Использование динамической библиотеки:

Создание .so (shared library) или .dll (Windows) файла для функций шифрования и дешифровки

4. Функции стандартной библиотеки C:

- `printf` и `fprintf` для вывода данных
- `argc` и `argv` для доступа к аргументам командной строки
- `EXIT_SUCCESS` и `EXIT_FAILURE` для обозначения успешного или неудачного завершения программы

Задание

Вариант 27

Счастливый билет

Задание

«Счастливым» называется билет, в котором в шестизначном номере сумма первых трех цифр равна сумме последних трех. Решение строится на основе полного перебора всех шестизначных чисел. Каждое из них следует разложить на цифры, а затем сравнить суммы первых и последних трех. Как видим, решение складывается из стандартных фрагментов, оформленных в виде функций, нужно только выложить их в нужной последовательности «сверху вниз».

Пояснительная записка:

1. Файл main.c

1. Подключение библиотек:

```
1. #include <stdio.h>
2. #include <stdlib.h>
3. #include <string.h>
```

Здесь подключаются стандартные библиотеки C:

- `stdio.h` для работы с вводом и выводом.
- `stdlib.h` для доступа к функциям стандартной библиотеки, таким как `malloc` и `exit`.
- `string.h` для функций обработки строк, таких как `strcmp`.

2. Функция проверки счастливого билета

```
int is_lucky_ticket(int number) {
    int number_copy = number;
    int digits[6];
    for (int i = 0; i < 6; i++) {
        digits[i] = number_copy % 10;
        number_copy /= 10;
    }
}
```

```

int sum1 = digits[0] + digits[1] + digits[2];
int sum2 = digits[3] + digits[4] + digits[5];

return sum1 == sum2;
}

```

- Функция `is_lucky_ticket` принимает номер билета в качестве параметра.
- Она создает указатель `number_cору`, который будет использоваться для разложения номера на цифры.
- Цикл `for` разлагает номер на цифры и сохраняет их в массиве `digits`.
- Вычисляются суммы первых трех и последних трех цифр.
- Функция возвращает 1, если суммы равны (билет счастливый), и 0 в противном случае.

3. Функция `main`

```

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

    if (argc < 3) {
        printf("Недостаточно аргументов командной строки. Использование: %s <номер билета 1> <имя_файла>\n", argv[0]);
        return EXIT_FAILURE;
    }

    FILE *fp = fopen(argv[argc - 1], "w");
    if (fp == NULL) {
        printf("Не удалось открыть файл %s для записи.\n", argv[argc - 1]);
        return EXIT_FAILURE;
    }

    for (int i = 1; i < argc - 1; i++) {
        int number;
        if (sscanf(argv[i], "%d", &number) != 1) {
            printf("Не удалось преобразовать аргумент %d в число.\n", i);
            fclose(fp);
            return EXIT_FAILURE;
        }

        if (number < 100000 || number > 999999) {
            printf("Число %d не является шестизначным.\n", number);
            fclose(fp);
            return EXIT_FAILURE;
        }
    }
}

```

```

    int result = is_lucky_ticket(number);
    if (result) {
        fprintf(fp, "Билет %d счастливый.\n", number);
    } else {
        fprintf(fp, "Билет %d не счастливый.\n", number);
    }
}

fclose(fp);

return EXIT_SUCCESS;
}

```

- Функция main проверяет количество аргументов командной строки. Если их меньше трех, выводится сообщение об ошибке и программа завершается.
- Открывается текстовый файл с именем, указанным в последнем аргументе командной строки. Если файл не удастся открыть, выводится сообщение об ошибке и программа завершается.
- Для каждого номера билета, указанного в аргументе командной строки, он извлекается и преобразуется в число. Если преобразование не удастся, выводится сообщение об ошибке и программа завершается.
- Проверяется, является ли номер шестизначным. Если нет, выводится сообщение об ошибке и программа завершается.
- Функция is_lucky_ticket вызывается для каждого номера, и результат записывается в текстовый файл.
- Файл закрывается, и программа завершается, возвращая EXIT_SUCCESS.

Псевдокод:

1.main.c

```
#include "head.h"

int main(int argc, char *argv[]) {
    // Проверка количества аргументов командной строки
    if (argc < 3) {
        printf("Недостаточно аргументов командной строки. Использование: %s <номер билета 1> <имя_файла>\n", argv[0]);
        return EXIT_FAILURE;
    }

    // Открытие текстового файла для записи
    FILE *fp = fopen(argv[argc - 1], "w");
    if (fp == NULL) {
        printf("Не удалось открыть файл %s для записи.\n", argv[argc - 1]);
        return EXIT_FAILURE;
    }

    // Обработка каждого номера билета
    for (int i = 1; i < argc - 1; i++) {
        // Преобразование аргумента командной строки в число
        int number;
        if (sscanf(argv[i], "%d", &number) != 1) {
            printf("Не удалось преобразовать аргумент %d в число.\n", i);
            fclose(fp);
            return EXIT_FAILURE;
        }

        // Проверка, является ли число шестизначным
        if (number < 100000 || number > 999999) {
            printf("Число %d не является шестизначным.\n", number);
            fclose(fp);
            return EXIT_FAILURE;
        }

        // Проверка, является ли билет счастливым
        int result = is_lucky_ticket(number);
        if (result) {
            fprintf(fp, "Билет %d счастливый.\n", number);
        } else {
            fprintf(fp, "Билет %d не счастливый.\n", number);
        }
    }

    fclose(fp);

    return EXIT_SUCCESS;
}
```

2.fsb.c

```
#include "head.h"
int is_lucky_ticket(int number) {
    // Инициализация указателя number
    int number_copy = number;

    // Разложение числа на цифры
    int digits[6];
    for (int i = 0; i < 6; i++) {
        digits[i] = number_copy % 10;
        number_copy /= 10;
    }

    // Вычисление сумм первых и последних трех цифр
    int sum1 = digits[0] + digits[1] + digits[2];
    int sum2 = digits[3] + digits[4] + digits[5];

    // Проверка, является ли билет счастливым
    return sum1 == sum2;
}
```

3.head.h

```
#include <stdio.h>
#include <stdlib.h>
#include <string.h>
int is_lucky_ticket(int number);
```


Компиляция и запуск:

1. Сохраните файлы:

- main.c
- fsb.c
- head.h

2.Объектный файл с главной функцией:

gcc -c main.c

3.Объектный файл для создания дин. либы:

gcc -c -fPIC fsb.c

4.Собираем дин. библиотеку:

gcc -shared -o libMY_PRINT.dll fsb.o

5.Линковка:

gcc -o dyn_example main.o -L. -lMY_PRINT

6.Запуск

./dyn_example

Пример вывода:

```
PS C:\work\Prog\rgz> gcc -c main.c
PS C:\work\Prog\rgz> gcc -c -fPIC fsb.c
PS C:\work\Prog\rgz> gcc -shared -o libMY_PRINT.dll fsb.o
PS C:\work\Prog\rgz> gcc -o dyn_example main.o -L. -lMY_PRINT
PS C:\work\Prog\rgz> ./dyn_example 234567 122345 876678 987654 123123 rs.txt
PS C:\work\Prog\rgz> █
```

rs.txt

1 Билет 234567 не счастливый.

2 Билет 122345 не счастливый.

3 Билет 876678 счастливый.

4 Билет 987654 не счастливый.

5 Билет 123123 счастливый.

6