

# Отчет по лабораторной работе № 12 по курсу “Фундаментальная информатика”

Студент группы М80-109Б-22 Фомин Иван Дмитриевич, № 22

Контакты email: [grenka388@gmail.com](mailto:grenka388@gmail.com),

Telegram: @Haliaven

Работа выполнена: «26» ноября 2022г.

Преподаватель: каф. 806 Сысоев Максим Алексеевич

Отчет сдан «28» \_\_11\_\_ 2022 г., итоговая оценка \_\_\_\_

Подпись преподавателя \_\_\_\_\_

1. **Тема:** Техника работы с целыми числами. Системы счисления.
2. **Цель работы:** Составить программу на языке Си в целом типе данных, которая для любых допустимых и корректно записанных чисел выполняет указанное действие.
3. **Задание:** Вариант 31. Упорядочить цифры числа попарно по убыванию.
4. **Оборудование** (студента):

Процессор AMD Ryzen 7 4800HS with Radeon Graphics с ОП 8 Гб, SSD 512 Гб. Монитор 1920x1080

5. **Программное обеспечение** (студента):

Операционная система семейства: *linux*, наименование: *ubuntu*, версия 22.04 *jammy*

интерпретатор команд: *bash* версия 5.1.16(1)-release

Система программирования CLion версия 2021.1.3, редактор текстов *nano* версия 6.2

Утилиты операционной системы WinRar, Microsoft Word

Прикладные системы и программы Ubuntu wsl, CLion, Google Chrome

Местонахождение и имена файлов программ и данных на домашнем компьютере ~/home

## 6. Идея, метод, алгоритм

Число можно реконструировать, если знать, на какой позиции стояла определённая цифра, тогда получится следующее:

$$A * 10^0 + B * 10^1 + C * 10^2 + \dots +$$

где A, B, C – цифры числа

A упорядочить цифры в паре можно путём обычного оператора >|<.

## 7. Сценарий выполнения работы

Входные данные	Выходные данные	Описание тестируемого случая
1	1	Число длиной 1
1 2 3	1 2 3	Несколько чисел длиной 1
123456	214365	Обычный случай
1234567	2143657	Нечётное количество цифр

## 8. Распечатка протокола

```
#include <stdio.h>
#include <ctype.h>
#include <stdint.h>
#include <assert.h>

int64_t powFunc(int64_t num, int64_t power) {
    if (power == 0) return 1;

    int64_t res = 1;

    while (power-- > 0)
        res *= num;

    return res;
}

int64_t getNewPair(char digit1, char digit2, int64_t pairsDone, int isReversed) {
    if (isReversed == 1 ? digit1 >= digit2 : digit1 <= digit2)
        return digit2 * powFunc(10, pairsDone * 2) + digit1 * powFunc(10,
pairsDone * 2 + 1);
    else return digit1 * powFunc(10, pairsDone * 2) + digit2 * powFunc(10,
pairsDone * 2 + 1);
}

int64_t reverseNum(int64_t num, int64_t pairsDone, char isOdd) {
    int64_t counter = isOdd == 1 ? 0 : -1;
    int64_t res = 0;

    while (num > 0) {
        res += num % 10 * powFunc(10, pairsDone * 2 - counter++);
        num /= 10;
    }

    return res;
}

void test_powFunc() {
    assert(powFunc(10, 0) == 1);
    assert(powFunc(1, 100) == 1);
    assert(powFunc(10, 2) == 100);
    assert(powFunc(2, 9) == 512);
    assert(powFunc(0, 9) == 0);
}

void test_getNewPair() {
    assert(getNewPair('4' - '0', '1' - '0', 0, 0) == 14);
    assert(getNewPair('1' - '0', '4' - '0', 0, 0) == 14);
    assert(getNewPair('9' - '0', '9' - '0', 0, 0) == 99);
}

void test_reverseNum() {
    assert(reverseNum(123456, 2, 0) == 654321);
    assert(reverseNum(654321, 2, 0) == 123456);
    assert(reverseNum(5454545, 3, 1) == 5454545);
}

void execTests() {
    test_powFunc();
    test_getNewPair();
    test_reverseNum();
}
```

```

int main() {
    execTests();

    // попарно по убыванию
    // 1234
    // 1 * 10**3 + 2 * 10**2 + 3 * 10**1 + 4 * 10**0
    char c;

    char dig1 = -1;
    char dig2 = -1;
    char isOdd = 0;
    int64_t resNum = 0;
    int64_t pairsDone = 0;

    while ((c = getchar()) != 'a') {

        if (isdigit(c) == 0){
            if (resNum == 0) {
                if (dig1 == -1 && dig2 == -1) continue;
                printf("%d ", dig1 != -1 ? dig1 : dig2);

                dig1 = -1;
                dig2 = -1;
                continue;
            }

            if (dig1 != -1 || dig2 != -1) {
                if (dig1 != -1 && dig2 == -1) {
                    resNum += dig1 * powFunc(10, pairsDone * 2);
                    isOdd = 1;
                } else resNum += getNewPair(dig1, dig2, pairsDone, 0);
            }
            printf("%lld ", reverseNum(resNum, pairsDone, isOdd));
            resNum = 0;

            dig1 = -1;
            dig2 = -1;
            isOdd = 0;
            pairsDone = 0;

            continue;
        }

        if (dig1 == -1 || dig2 == -1) {
            if (dig1 == -1) dig1 = c - '0';
            else dig2 = c - '0';
        } else {
            resNum += getNewPair(dig1, dig2, pairsDone, 0);

            dig1 = c - '0';
            dig2 = -1;

            pairsDone++;
        }
    }

    if (dig1 != -1 && dig2 != -1) {
        resNum += getNewPair(dig1, dig2, pairsDone, 0);
        printf("%lld", resNum);
    } else if (dig1 != -1) printf("%d", dig1);
    else if (dig2 != -1) printf("%d", dig2);
}

```

## 9. Дневник отладки

№	Лаб. или дом.	Дата	Время	Событие	Действие по исправлению	Примечание
1	дом	25.11.2022	19:00	Каждый раз программа почему-то заменяла всё, что не является цифрами, на нули	Добавил if в if (isDigit(a))	И опять компилятор никак не помог

## 10. Замечания автора

Всё норм, но если прямо придирается, то можно сказать, что для подобного взаимодействия с числами лучше всего использовать массивы или переводить число в строку (что как бы тоже является массивом), а моё решение уже не очень современное

## 11. Выводы

Пока могу сказать, что работа в разы приятнее 11 лабы, просто потому, что тут я не упирался в потолок из-за того, что чего-то не знаю. Сама работа с числами более-менее схожа с другими ЯП, поэтому в принципе было не так сложно. Однако это всё же опыт программирования на Си, а это значит, что в дальнейшем будет проще (хотя бы чуть-чуть) писать программы на Си.

Недочёты при выполнении задания могут быть устранены следующим образом: --

Подпись студента \_\_\_\_\_