

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

Студент группы М80-109Б-22 Федоров Алексей Алексеевич, № 20

Контакты:

Email: hedgefog@yandex.ru, Telegram: @hedgefo9

Работа выполнена: «05» декабря 2022г.

Преподаватель: каф. 806 Сысоев Максим Алексеевич

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

Подпись преподавателя

1. **Тема:** Техника работы с целыми числами. Системы счисления.

2. **Цель работы:** Научиться работать с целыми числами.

3. **Задание (Вариант №10)**

Выполнить циклический сдвиг влево десятичного представления числа.

4. **Оборудование (студента):**

Процессор *Ryzen 5 3500U @ 8x 2.1GH* с ОП *14900* Мб, НМД *677* Гб. Монитор *2160x1440*

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

Операционная система семейства: *linux*, наименование: *ubuntu*, версия *18.10 cinnamon* интерпретатор команд: *bash* версия *4.4.19*.

Система программирования -- версия --, редактор текстов *emacs* версия *25.2.2*

6. **Идея, метод, алгоритм**

Идея заключается в том, чтобы вычислять Y , равное 10 в степени (количество разрядов у числа – 1). Таким образом, мы можем получить первую цифру числа, поделив число на Y , а также все остальные цифры числа, взяв остаток от деления числа на Y . В итоге для получения циклически сдвинутого влево числа нужно умножить все остальные цифры на 10 и прибавить старую первую цифру. Чтобы циклически сдвинуть число на N , нужно сдвинуть N раз число на 1.

Формат входных данных:

Строка 1 – число, которое нужно сдвинуть.

Строка 2 – число, на которое нужно сдвинуть влево число в Строке 1.

7. **Сценарий выполнения работы**

Unit-тесты проведены внутри программы, см. п. 8

8. **Распечатка протокола:**

Код на языке C

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

#define INT_MIN (-2147483648)
#define INT_MAX 2147483647

// возвращает модуль числа
int abs(int a) {
    if (a < 0) {
        return -a;
    } else {
        return a;
    }
}

// возвращает знак числа
int sign(int a) {
    if (a > 0) {
        return 1;
    } else if (a == 0) {
        return 0;
    } else if (a < 0) {
        return -1;
    }
}
```

```

// возвращает количество разрядов у числа
int digits_quantity(int x) {
    int x_abs = abs(x);
    if (x_abs == 0) {
        return 1;
    }
    int num_len = 0;
    while (x_abs > 0) {
        num_len++;
        x_abs = x_abs / 10;
    }
    return num_len;
}

// возвращает число, равное x возведённому в положительную или нулевую степень power
int pow_int(int x, int power) {
    int res = 1;
    for (int i = 0; i < power; i++) {
        res *= x;
    }
    return res;
}

// возвращает число, равное сдвинутому числу x на shift_val влево
int shift_left(int x, int shift_val) {
    int new_num = x;
    int x_len = digits_quantity(x);
    shift_val = shift_val % x_len;
    int y = pow_int(10, x_len - 1);
    for (int k = 0; k < shift_val; k++) {
        int x_tmp = new_num;
        new_num = (x_tmp % y) * 10 + (x_tmp / y);
    }
    return new_num;
}

void test_abs() {
    assert(abs(5) == 5);
    assert(abs(-3) == 3);
    assert(abs(0) == 0);
}

void test_sign() {
    assert(sign(7) == 1);
    assert(sign(-10) == -1);
    assert(sign(0) == 0);
}

void test_digits_quantity() {
    assert(digits_quantity(0) == 1);
    assert(digits_quantity(1234) == 4);
    assert(digits_quantity(348584739) == 9);
    assert(digits_quantity(-10000) == 5);
}

void test_pow_int() {
    assert(pow_int(10, 4) == 10000);
    assert(pow_int(2, 20) == 1048576);
    assert(pow_int(100, 0) == 1);
    assert(pow_int(-5, 2) == 25);
    assert(pow_int(-7, 5) == -16807);
}

void test_shift_left() {
    assert(shift_left(1234, 1) == 2341);
    assert(shift_left(230503604, 2) == 50360423);
    assert(shift_left(9284158, 5764801) == 9284158);
}

```

```

void test_all() {
    test_abs();
    test_sign();
    test_digits_quantity();
    test_pow_int();
    test_shift_left();
}

int main() {
    test_all();

    int num;
    int shift_value;
    scanf("%d\n", &num);
    scanf("%d", &shift_value);

    if (num == 0) {
        printf("%d", 0);
        return 0;
    }

    // сохранение знака
    int sign_original = sign(num);
    int num_abs = abs(num);

    int shifted_num = shift_left(num_abs, shift_value) * sign_original;
    printf("%d\n", shifted_num);

    return 0;
}

```

9. Дневник отладки должен содержать дату и время сеансов отладки и основные события (ошибки в сценарии и программе, нестандартные ситуации) и краткие комментарии к ним. В дневнике отладки приводятся сведения об использовании других ЭВМ, существенном участии преподавателя и других лиц в написании и отладке программы.

№	Лаб. или дом.	Дата	Время	Событие	Действие по исправлению	Примечание
1	Лаб	05.12.2022	12:00	Я устал писать unit-тесты	Отдохнул.	Всё хорошо.

10. Замечания автора

Нет замечаний

11. Выводы

Благодаря этой лабораторной работе я потренировался работать с числами. Ничего нового не узнал, но зато попрактиковался писать Unit-тесты, что в целом интересно.

Подпись студента _____