

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 (вправо) 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 представлять число в виде массива цифр. Таким образом, мы можем циклически сдвигать массив (вправо или влево), а в итоге преобразовывать его в число. Чтобы циклически сдвинуть число на N, нужно сдвинуть N раз массив цифр на 1.

Формат входных данных:

Строка 1 – число, которое нужно сдвинуть.

Строка 2 – символ «l» (если нужно сдвинуть число влево) или «r» (если нужно сдвинуть число вправо)

Строка 3 – число, на которое нужно сдвинуть число в Строке 1

7. **Сценарий выполнения работы**

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

8. **Распечатка протокола:**

```
#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 сдвинутое десятичное число на shift_val вправо
```

```

int shift_right(int digits[10], int num_len, int shift_val) {
    shift_val = shift_val % num_len;
    for (int k = 0; k < shift_val; k++) {
        int new_first_digit = digits[num_len - 1];
        for (int i = num_len - 1; i > 0; i--) {
            digits[i] = digits[i - 1];
        }
        digits[0] = new_first_digit;
    }
    int new_num = 0;
    for (int i = 0; i < num_len; i++) {
        new_num = new_num * 10 + digits[i];
    }
    return new_num;
}

// возвращает сдвинутое десятичное число на shift_val влево
int shift_left(int digits[10], int num_len, int shift_val) {
    shift_val = shift_val % num_len;
    for (int k = 0; k < shift_val; k++) {
        int new_last_digit = digits[0];
        for (int i = 0; i < num_len - 1; i++) {
            digits[i] = digits[i + 1];
        }
        digits[num_len - 1] = new_last_digit;
    }
    int new_num = 0;
    for (int i = 0; i < num_len; i++) {
        new_num = new_num * 10 + digits[i];
    }
    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_shift_right() {
    int digits_test_1[10] = {1, 2, 3, 4, 0, 0, 0, 0, 0, 0};
    assert(shift_right(digits_test_1, 4, 1) == 4123);
    int digits_test_2[10] = {3, 0, 5, 6, 4, 8, 9, 0, 0, 0};
    assert(shift_right(digits_test_2, 7, 3) == 4893056);
    int digits_test_3[10] = {7, 8, 3, 2, 0, 0, 0, 1, 1, 0};
    assert(shift_right(digits_test_3, 9, 10) == 178320001);
}

void test_shift_left() {
    int digits_test_1[10] = {1, 2, 3, 4, 0, 0, 0, 0, 0, 0};
    assert(shift_left(digits_test_1, 4, 1) == 2341);
    int digits_test_2[10] = {3, 0, 5, 6, 4, 8, 9, 0, 0, 0};
    assert(shift_left(digits_test_2, 7, 3) == 6489305);
    int digits_test_3[10] = {7, 8, 3, 2, 0, 0, 0, 1, 1, 0};
    assert(shift_left(digits_test_3, 9, 10) == 832000117);
}

void test_all() {
    test_abs();
    test_sign();
    test_shift_right();
    test_shift_left();
}

```

```

int main() {
    test_all();

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

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

    int sign_original = sign(num);
    int num_abs = abs(num);

    int digits[10];
    for (int i = 0; i < 10; i++) digits[i] = 0;

    int num_len = 0;
    int num_tmp = num_abs;

    // преобразования числа в массив с цифрами
    while (num_tmp > 0) {
        int current_digit = num_tmp % 10;
        digits[num_len] = current_digit;
        num_len++;
        num_tmp = num_tmp / 10;
    }

    // реверс числа внутри массива
    for (int i = 0; i < num_len / 2; i++) {
        int tmp = digits[i];
        digits[i] = digits[num_len - i - 1];
        digits[num_len - i - 1] = tmp;
    }
    int shifted_num = 0;
    switch (direction) {
        case 'l':
            shifted_num = shift_left(digits, num_len, shift_value) * sign_original;
            break;
        case 'r':
            shifted_num = shift_right(digits, num_len, shift_value) * sign_original;
            break;
        default:
            printf("%s", "Непонятное направление, на этом наши полномочия всё\n");
            break;
    }
    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 _____