

Отчет по лабораторной работе №12 по курсу Фундаментальная информатика

Студент группы М8О-104Б-22 Ляпин Иван Алексеевич, № по списку 00

Контакты www, e-mail, icq, skype shad0w2020@mail.ru

Работа выполнена: « 5 » ноября 2022 г.

Преподаватель: асп. каф. 806 Потенко М.А.

Входной контроль знаний с оценкой _____

Отчет сдан « » _____ 202 __ 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 в десятичном изображении выполняет указанное вариантом действие.

3. **Задание (вариант № 27):** После каждых двух цифр записать абсолютное значение их разности.

4. **Оборудование (лабораторное):**

ЭВМ _____, процессор _____, имя узла сети _____ с ОП _____ Мб,
НМД _____ Мб. Терминал _____ адрес _____. Принтер _____
Другие устройства _____

Оборудование ПЭВМ студента, если использовалось:

Процессор Apple M1 Pro с ОП 16384 Мб, НМД 524 288 Мб. Монитор Liquid Retina XDR

Другие устройства _____

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

Операционная система семейства _____, наименование _____ версия _____

интерпретатор команд _____ версия _____

Система программирования _____ версия _____

Редактор текстов _____ версия _____

Утилиты операционной системы _____

Прикладные системы и программы _____

Местонахождение и имена файлов программ и данных _____

Программное обеспечение ЭВМ студента, если использовалось:

Операционная система семейства macOS, наименование macOS Monterey версия 12.3

интерпретатор команд zsh версия 2.12.5

Система программирования C версия _____

Редактор текстов nano версия _____

Утилиты операционной системы Терминал _____

Прикладные системы и программы Xcode

Местонахождение и имена файлов программ и данных /Users/ivan/Desktop _____

6. Идея, метод, алгоритм решение задачи (в формах: словесной, псевдокода, графической [блок-схема, диаграмма, рисунок, таблица] или формальные спецификации с пред- и постусловиями)

Будем обрабатывать входное число справа-налево для удобства.

Так как нам нужно исследовать пары цифр в числе, то необходимо учесть четность и нечетность длины числа, первоначально вычислить эту длину.

Так как наша программа должна работать для чисел любого знака, то нужно ввести переменную(is minus), принимающую значение (1, -1). Если наше число < 0 , то переменная будет иметь отрицательное значение.

Так как наша программа должна выполняться, пока мы сами не прекратим работу, то очевидно, что считывание вводимого числа должно происходить в цикле, пока не встретим EOF.

Напишем две функции для последующих операций над числом:

even - если длина числа четная.

odd - если длина числа нечетная.

Вызовем соответствующую функцию, и выведем произведение значения функции и параметра "is minus" в результат.

Далее обнулим значение длины числа и значения "is.minus" для дальнейших операций с новыми числами

even(вводимое число):

Выполним проверку на переполнение типа int(если число больше INT_MAX, то выведем функцией уведомление о переполнении.)

Будем работать в цикле, пока наше число > 0 . Так как нужно просматривать пары цифр, то каждую цифру пары будем запоминать с помощью операции вычисления остатка на 10, так как число десятичное, и чтобы избавиться от только что вычисленного числа будем делить число на 10.

Далее будем вычислять разницу между первым и вторым числом.

Затем нужно составить новое число, путем умножения первого числа на некоторый параметр(transition), изначально равный 1, далее прибавим второе число, умноженное на 10 и transition, после добавим значение разницы, умноженное на 100 и transition.

Так как нужно перебрать все пары числа, то очевидно, что необходимо умножать параметр transition на константу MOVE, равную 1000.

После выполнения цикла, выводим результат.

odd(вводимое число):

Выполняем идентичные действия как у функции even, но перед выводом результата, необходимо взять остаток(10) от вводимого число, умножить его на значение transition после цикла и прибавить к числу-результату цикла.

выводим результат.

7. Сценарий выполнения работы (план работы, первоначальный текст программы в черновике [можно на отдельном листе] и тесты либо соображения по тестированию)

1) Введем переменные: num_keyboard(входное число с клавиатуры), num_k-check(будем вычислять длину числа, благодаря этой переменной), length(длина числа), is_minus(bool значение, опр. знак числа "+" или "-"), sign(знак числа, зависящий от значения is_minus)

2) Запустим цикл, работающий пока не захотим его прервать с помощью "D".

3) Для начала проверим знак входного числа(если число < 0, то is_minus = true и sign = -1)

4) Приравниваем num_k-check к num_keyboard и запускаем цикл с условием, пока num_k-check != 0. В цикле увеличиваем значение length на единицу, затем делим число на 10, таким образом вычисляем длину числа.

5) Если длина четная, то выводим результат произведения функции even(num_keyboard) * sign. Если длина нечетная - результат произведения функции odd(num_keyboard) * sign.

6) Обнуляем значение length и присваиваем значение "false" переменной is_minus для дальнейшей работы в цикле.

even(num_keyboard):

updated num - значение функции, то есть число, что получится на выходе.

transition - счетчик перехода от одной пары чисел к последующей

first, second, third - переменные пары(first, second) и значение разности(third)

Проверяем, является ли входное число числом, большим INT-MAX (если да, то выводим уведомление о переполнении типа int и завершаем работу функции)

Запускаем цикл, пока входное число > 0, где будем работать над парами чисел:

считываем число(присваиваем к first) путем вычисления остатка при дел. на 10, затем удаляем это число делением нацело на 10, повторяем операции и присваиваем к "second".

Вычисляем абсолютную разность и записываем значение к переменной "third".

Вычисляем updated num путем суммы произведений (first * transition, second * 10 * transition, third * 100 * transition)

увеличиваем значение "transition" в 1000(const MOVE) раз для перехода к следующей паре чисел.

Выводим результат

odd(num_keyboard):

Основная часть идентична работе функции even, но перед выводом числа, следует добавить к нему значение переменной "last = num_keyboard%10", умноженной на значение transition после выполнения цикла.

Выводим результат.

Пункты 1-7 отчета составляются строго до начала лабораторной работы.

Допущен к выполнению работы. Подпись преподавателя _____

преподавателем)

Код

```

Данил Иван Алексеевич
МВО-1046-22
Лабораторная работа #12
Вариант #27
*/

#include <math.h>
#include <ctype.h>
#include <stdio.h>
#include <stdlib.h>
#include <string.h>
#include <limits.h>
#include <stdbool.h>

#define MOVE 1000

long long even (long long a) // если длина введенного числа четная
{
    unsigned long long updated_num = 0; //значение функции
    long long transition = 1; // создаем счетчик для перехода к следующей паре цифр

    int first, second, third; //first - последняя цифра числа, second - предпоследняя,
    //third - их разность под модулем, transition - коэффициент для записи выходного числа new
    if (a > INT_MAX) { // проверка на переполнение int
        printf("Переполнение!Введите новое число");
        return 0;
    }
    while (a > 0) {
        first = a % 10; // берем последний разряд числа и запоминаем
        a /= 10; // "вычеркиваем" его
        second = a % 10; // берем следующий разряд и запоминаем
        a /= 10; // "вычеркиваем" его
        third = abs(first - second); // вычисляем модуль разности
        /* восстанавливаем исходное число по паре + значение их разности */
        updated_num += first * transition + second * 10 * transition + third * 100 * transition;
        transition *= MOVE; //увеличиваем transition, так как мы предыдущие разряды уже записали
    }
    return updated_num;
}

long long odd (long long b) // если длина введенного числа нечетная
{
    // делаем аналогичные функции even()
    unsigned long long updated_num = 0;

    long long transition = 1;

    int first, second, third, last; // last - цифра, что будет добавлено в начало обновленного числа, так как длина входного числа нечетная, а у ч

    if (b > INT_MAX) {
        printf("Переполнение!Введите новое число");
        return 0;
    }
    while (b >= 10) {
        first = b % 10;
        b /= 10;
        second = b % 10;
        b /= 10;
        third = abs( first - second);
        updated_num += first * transition + second * 10 * transition + third * 100 * transition;
        transition *= MOVE;
    }
    last = b % 10;
    updated_num += last * transition; // добавляем последнюю цифру к new, так как у этой цифры нет пары для сравнения
    return updated_num;
}

int main()
{
    long long num_keyboard, num_k_check; // num_keyboard - число, введенное с клавиатуры, вводим переменную num_k_check для вычисления длины числа
    int length = 0; // длина числа
    bool is_minus = false; //is_minus - для определения знака
    int sign = 1;
    printf("Введите число или '0' для завершения программы: \n");
    while (scanf("%lld", &num_keyboard) != EOF) {
        if (num_keyboard < 0) { // проверка на знак
            num_keyboard = abs( num_keyboard );
            is_minus = true;
            sign = -1;
        }
        num_k_check = num_keyboard;
        while (num_k_check != 0 ) { // проверяем на четность и нечетность длину нашего числа
            length++;
            num_k_check /= 10;
        }
        if (length % 2 == 0) { //если длина числа четная, то вызываем соответствующую функции even
            printf("К1d\n", even(num_keyboard) * sign); // функция обработки четного числа
        }
        else { //если нечетная длина, то вызываем соответствующую функции odd
            printf("К1d\n", odd(num_keyboard) * sign); // функция обработки нечетного числа
        }
        length = 0; // возвращаем наши значения для дальнейших операций
        is_minus = false;
        printf("Введите число или '0' для завершения программы: \n");
    }
}

```

Тесты

```
Введите число или ^D для завершения программы:
6765734
6176257134
```

```
Введите число и  $\Delta D$  для завершения программы:
2147483640
121347448336440
```

```
Введите число или ^D для завершения программы:  
99999999999999999999  
Переполнение! Введите новое число0
```

```
Введите число или ^D для завершения программы:
121212120
11211121121220
```

```
Введите число или ^D для завершения программы
90909
9909909
```

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 ЭВМ, существенном участии преподавателя и других лиц в написании и отладке программы.

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

10. **Замечания автора** по существу работы: В данной работе обязательным условием является выполнение вычислений в рамках границы типа переменной int. Так как мы можем использовать значения, близкие к INT_MAX, то обязательным условием была инициализация функций even, odd как unsigned long long.

11. Выводы: В данной лабораторной работе я научился работать с целыми числами, совершать различные операции над ними. Научился писать более сложные функции чем раньше, а также работать с диапазоном различных типов переменных, таких как int, long long, а также unsigned long long

Недочёты при выполнении задания могут быть устранены следующим образом:

Подпись студента

A handwritten signature in black ink, consisting of a stylized 'L' shape with a horizontal line extending to the right and a small loop at the bottom.