

Отчет по лабораторной работе №12 по курсу фундаментальная информатика

Студент группы М8О-104Б-22 Ольхов Максим Алексеевич, № по списку 11

Контакты www, e-mail, icq, skype maksolhov@yandex.ru

Работа выполнена: «8» октября 2022г.

Преподаватель: Потенко М.А. каф. 806

Входной контроль знаний с оценкой _____

Отчет сдан « » _____ 201 ____ 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 № 2*):** выбрать тройки подряд идущих цифр числа, сумма которых максимальна.
4. **Оборудование (лабораторное):**
ЭВМ _____, процессор _____, имя узла сети _____ с ОП _____ Мб, НМД _____ Мб. Терминал _____ адрес _____
_____. Принтер _____
Другие устройства _____
Оборудование ПЭВМ студента, если использовалось:
Процессор Intel core i5-7500 3.4 GHz с ОП 8 Гб, НМД 2097152 Мб. Монитор asus
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 windows, наименование windows 10 версия 10
интерпретатор команд _____ версия _____
Система программирования _____ версия _____ Эмулятор машины Тьюринга в четвёрках v2.3 _____
Редактор текстов _____ версия _____
Утилиты операционной системы _____ Яндекс _____
Прикладные системы и программы Яндекс
и имена файлов программ и данных на домашнем компьютере _____
/Users/max/test _____

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

Буду работать с числом справа-налево для более комфортной работы.

Заводим переменную `num` – это будет вводимое нами число, которое мы должны анализировать.

Так как наша программа должна работать с любым числом, которое больше 99, мы должны учесть это. Для этого я само число умножаю на “-1”. После чего работа программы будет корректна. Дальше мы должны учесть, что число должно быть больше 99, т.к программа должна работать с тройками цифр, а в числах меньше 100, нет 3 цифры. Мы просто пропускаем такое число. (!!Помечание: В данном отрывке идеи сначала лучше, конечно, пропустить число, которое будет меньше 100, чтобы лишний раз двухзначное число не умножать на “-1” и потом уже пропускать.)

Если нам число все-таки подходит, мы заводим цикл, который будет брать последнюю, предпоследнюю и пред предпоследнюю цифру, для этого нам нужно будет 3 переменные, которые будут содержать эти цифры. После чего мы уменьшаем разрядность самого числа делением на 10. Далее нам нужно будет считать сумму 3 цифры, которые мы запомнили в трех переменных, сумму будем запоминать в переменной `sum`. Для того, чтобы найти максимальную сумму, мы будем сравнивать переменной `max_sum`, которая будет равна -1. Если сумма 3 цифр будет больше переменной `max_sum`, то мы `sum` присваиваем к `max_sum`. Именно таким способом можно вычислить максимальную сумму. Если `sum` будет меньше `max_sum`, то цикл начнет работать с другими тремя цифрами. Цикл будет работать до тех пор, пока число не станет равным 0.

Для вывода цифр я завел еще три переменные, в которые буду запоминать те самые 3 цифры, которые дают максимальную сумму. После выполнения цикла и вывода 3 цифр с их суммой, я обнуляю переменную `max_sum`, чтобы подсчет максимальной суммы следующих чисел был корректен.

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

1) Заведём три константы, которые будут отвечать за разряд числа:

```
#define THE_NUMBER_OF_HUNDREDS 1000
```

```
#define THE_NUMBER_OF_TENS 100
```

```
#define THE_NUMBER_OF_UNITS 10
```

2) Далее у нас идет цикл, который будет работать до тех пор, пока мы сами этого не захотим, например, прервать его с помощью “^D”(CTRL+D).

3) Проверяем 2 условия:

Первое условие: Если “num<0”, т.е. число, которое мы ввели отрицательно, то мы должны умножить на -1.

Второе условие: Если “num<100”, т.е. число, которое мы ввели двухзначное, то мы не должны его анализировать, пропуская оператором “continue”.

4) Иначе мы заводим цикл, который будет работать, пока наш num!=0.

5) В начале этого цикла мы присваиваем 3 последние цифры к переменным (digit_last = num % THE_NUMBER_OF_UNITS, digit_pre_last = num % THE_NUMBER_OF_TENS / 10, digit_prepred_last = num % THE_NUMBER_OF_HUNDREDS / 100)

6) После того как присвоили 3 последние цифры, мы можем смело уменьшить разрядность числа путем делением на 10 (num /= 10).

7) Дальше у нас действует переменная sum, которая будет считать сумму 3 последних цифр (sum = digit_last + digit_pre_last + digit_prepred_last)

8) Для нахождения максимальной суммы, мы проверяем условие, если sum > max_sum (max_sum изначально равна -1).

9) Если условие выполняется, то мы нашли максимальную сумму 3 цифр(max_sum=sum). Далее мы у нас будут еще три переменные, в которые мы засунем 3 максимальные цифры для вывода (max_digit_prepred_last = digit_prepred_last, max_digit_pre_last = digit_pre_last, max_digit_last = digit_last).

10) Если условие не подходит, то мы просто продолжаем искать максимальную сумму 3 цифры, проходя всё число.

11) Как число станет равным нулю, мы выводим 3 цифры и их сумму, и после вывода обнуляем максимальную сумму(max_sum=0).

Тесты:

```
12345
тройки цифр 3 4 5 и их сумма 12
92658
тройки цифр 6 5 8 и их сумма 19
123
тройки цифр 1 2 3 и их сумма 6
-12324
тройки цифр 3 2 4 и их сумма 9
12
0
123 125 0 12
тройки цифр 1 2 3 и их сумма 6
тройки цифр 1 2 5 и их сумма 8
```

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

Допущен к выполнению работы. Подпись преподавателя _____

8. Распечатка протокола (подклеить листинг окончательного варианта программы с тестовыми примерами, подписанный преподавателем)

```
//Ольхов Максим Алексеевич
//лаба 12
//Вар 2
#include <stdio.h>
#define THE_NUMBER_OF_HUNDREDS 1000
#define THE_NUMBER_OF_TENS 100
#define THE_NUMBER_OF_UNITS 10

int main()
{
    int digit_last, digit_pre_last, digit_prepred_last, digit, num, sum = 0, max_sum = -1, max_digit_prepred_last, max_digit_pre_last, max_digit_last; //заводим переменные
    while (scanf("%d", &num) == 1) {
        if (num < 0) { //Если число меньше, то мы умножаем на -1, чтобы хорошо считалось считалось.
            num *= -1;
        }
        if (num < 100) { // Если число имеет меньше 3 цифр нам оно не нужно.
            continue;
        }
        else {
            while (num != 0) { // пока число не равно нулю, то мы проходим циклы, который будет идти по числу за каждой цифрой.
                digit_last = num % THE_NUMBER_OF_UNITS; // присваиваем последнюю цифру.
                digit_pre_last = num % THE_NUMBER_OF_TENS / 10; // присваиваем десяток.
                digit_prepred_last = num % THE_NUMBER_OF_HUNDREDS / 100; // присваиваем сотни.

                //уменьшаем разрядность числа.
                digit = num % 10;
                num /= 10;
                // сумма 3 цифр
                sum = digit_last + digit_pre_last + digit_prepred_last;
                if (sum > max_sum) { //Если сумма больше максимальной суммы, то мы присваиваем сумму к максимальной сумме и все цифры этой суммы.
                    max_sum = sum;
                    //присваиваем цифры из максимальной суммы, которая состоит из 3 цифр.
                    max_digit_prepred_last = digit_prepred_last;
                    max_digit_pre_last = digit_pre_last;
                    max_digit_last = digit_last;
                }
            }
            printf("тройки цифр %d %d %d и их сумма %d\n", max_digit_prepred_last, max_digit_pre_last, max_digit_last, max_sum);
            //обнуляем макс, чтобы анализ следующего числа был корректен.
            max_sum = 0;
        }
    }
    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

10. Замечания автора по существу работы:

11.Выводы

Данная работа дала возможность поближе познакомиться с целочисленными данными. Я более подробно проработал анализ десятичных чисел и совершать над ними какие-либо действия. Данные навыки необходимы для дальнейшего изучения языка Си.

Подпись студента _____