

Отчет по лабораторной работе № 8 по курсу “Фундаментальная информатика”

Студент группы М80-109Б-22 Любарский Иван Владимирович, № по списку 8

Контакты ivanred28@gmail.com, @rMEDGranD

Работа выполнена: «17» ноября 2022г.

Преподаватель: каф. 806 Сысоев Максим Алексеевич

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

Подпись преподавателя

- **Тема:** Системы программирования на языке Си.
- **Цель работы:** Разработать код на языке Си для решения поставленной задачи.
- **Задание:** Нахождение минимальной суммы, делящейся на 3. (Вариант 19)
- **Оборудование:**

Процессор Intel Core i5-4210U @ 4x 1.7GH с ОП 15873 Мб, НМД 512 Гб. МОНитор 1600x900

- **Программное обеспечение:**

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

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

Утилиты операционной системы --

Прикладные системы и программы --

Местонахождение и имена файлов программ и данных на домашнем компьютере --

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

Сумма из трех чисел делиться на три, если сумма их остатков от деления на 3 делиться на 3. В таком случае возможны такие комбинации остатков : 1+1+1, 0+1+2, 2+2+2.

Для определения минимальной суммы, делящейся на 3, требуется найти 3 минимальных числа для каждого остатка, которые дают остаток 2 и 1, а также минимальное, которое дает остаток 0.

В конце надо составить 3 суммы чисел, первая из остатков 1+1+1, вторая 0+1+2, третья 2+2+2. Надо сравнить все три суммы и вывести минимальную.

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

План выполнения кода:

1. Считать количество чисел в переменную N.
2. Создать два массива по 3 числа(для остатков 1 и 2) и переменную для 0.
3. Считать на вход число.
4. Если оно минимально для 2 или 1, то вставить в соответствующий массив, если минимально для 0, то вставить в соответствующую переменную.
5. Повторить действия 3-4 (N-1) раз.
6. Просуммировать числа с остатками 1+1+1, 0+1+2, 2+2+2.
7. Найти минимальную из трех сумм.
8. Вывести эту сумму.

Входные данные	Выходные данные	Описание тестируемого случая
4 5 8 11 14	24	Нахождения минимальной суммы для случая 2+2+2
6 3 4 5 8 11 14	12	Нахождения минимальной суммы для случая 0+1+2
7 8 3 11 4 14 1 4	9	Нахождения минимальной суммы для случая 1+1+1 с повторением числа и несортированными по порядку числами
8 5 8 3 11 4 14 1 7	9	Нахождения минимальной суммы для случая 0+1+2 с несортированными по порядку числами

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

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

```
int Min(int x, int y);
```

```
void FindMin(int number, int* mas);
```

```
int SumReminders(int* mas);
```

```
int main() {
```

```
    int count;
```

```
    scanf_s("%i", &count);
```

```
    int remainders2[3] = { 100000000, 100000000, 100000000 };
```

```
    int remainder0 = 100000000;
```

```
    int remainders1[3] = { 100000000, 100000000, 100000000 };
```

```
    int current_number;
```

```
    for (int i = 0; i < count; i++) {
```

```
        scanf_s("%i", &current_number);
```

```
        if (current_number % 3 == 0) { remainder0 = Min(current_number, remainder0); }
```

```
        if (current_number % 3 == 1) { FindMin(current_number, remainders1); }
```

```
        if (current_number % 3 == 2) { FindMin(current_number, remainders2); }
```

```
    }
```

```
    printf("%i", Min(Min(SumReminders(remainders1), (remainders1[0] + remainders2[0] + remainder0)),
```

```
SumReminders(remainders2)));
```

```
    return 0;
```

```
}
```

```
void FindMin(int number, int* mas) {
```

```

        if (number < mas[0]) { mas[2] = mas[1]; mas[1] = mas[0]; mas[0] = number; }
        else if (number < mas[1]) { mas[2] = mas[1]; mas[1] = number; }
        else if (number < mas[2]) { mas[2] = number; }
    }
    int Min(int x, int y) {
        if (x <= y) { return x; }
        else { return y; }
    }
    int SumReminders(int* mas){
        int sum = 0;
        for (int i = 0; i < 3; i++) { sum += mas[i]; }
        return sum;
    }

```

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.	Дата	Время	Событие	Действие по исправлению	Примечание
	или					
	дом.					
0						

10. Замечания автора по существу работы

11. Выводы

Данная работа с языком Си позволила вспомнить основы работы с Си подобными языками. Очень удобно писать код с возможностью создания переменных в памяти и функций, которые в несколько сот раз не только упрощают написание, но и понимание во время прочтения. Инструментарий чтения, вывода, хранения и преобразования данных делает язык Си базовым для работы в области программирования.

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

Подпись студента
