**C-lab-3**

**Лабораторная работа №3 (Операторы и выражения-2)**

**Задача №1**

Написать программу, подсчитывающую количество слов во

введённой пользователем строке

**Пояснение**

Строка, вводимая с клавиатуры может содержать любое количество пробельных и непробельных символов. Таким образом, мы рассматриваем слово как последовательность символов, разделенную любым количеством пробелов (пробелы могут находиться и в начале и в конце строки).

**Состав**

Программа должна состоять из двух функций:

- int wordCount(char buf[]) - подсчет количества слов

- main() - организация диалога.

Как и в задачах первого практикума создаются три файла: task1.h,task1.c,main1.c.

**Задача №2**

Написать программу, которая для введённой строки определяет ко-

личество слов и выводит каждое слово на отдельной строке и его

длину

**Пояснение**

Строка, вводимая с клавиатуры может содержать любое количество пробельных и непробельных символов. Таким образом, мы рассматриваем слово как последовательность символов, разделенную любым количеством пробелов (пробелы могут находиться и в начале и в конце строки).

**Состав**

Программа должна состоять из функции:

- main().

Текст программы помещается в файл main2.c

**Задача №3**

Написать программу, которая для введённой строки

выводит самое длинное слово его длину

**Пояснение**

Строка, вводимая с клавиатуры может содержать любое количество пробельных и непробельных символов. Таким образом, мы рассматриваем слово как последовательность символов, разделенную любым количеством пробелов (пробелы могут находиться и в начале и в конце строки).

**Состав**

Программа должна состоять из функций:

- int getMaxWord(char buf[],char word[]) - нахождение слова максимальной длины.

Слово записывается в массив word, а его длина возвращается из функции

- main().

Текст программы помещается в файлы: task3.h, task3.c, main3.c

**Задача №4**

Написать программу, которая находит сумму чисел во введённой строке

**Пояснение**

Программа рассматривает непрерывные последовательности цифр в строке как числа и обрабатывает их как единое целое. В программе предусмотреть ограни- чение на максимальное число разрядов, то есть если пользователь вводит очень длинную последовательность цифр, её нужно разбить на несколько групп.

*Пример:*

"ab12cd45e" -> 57

**Состав**

Программа должна состоять из функций:

- int getSum(char buf[]) - нахождение суммы

- main().

Текст программы помещается в файлы: task4.h, task4.c, main4.c

**Задача №5**

Написать программу, которая формирует целочисленный массив размера N,

а затем находит сумму элементов, расположенным между первым отрицательным

и последним положительным элементами.

**Пояснение**

Массив заполняется случайными числами: отрицательными и положительными поровну (или почти поровну...)

*Пример:*

[1 -2 **3 -4 -6 2** 3] -> -5

**Состав**

Программа должна состоять из функций:

- int getSumInt(int arr[],int N) - нахождение суммы в массиве arr длины N

- main().

Текст программы помещается в файлы: task5.h, task5.c, main5.c

**Задача №6**

Написать программу, которая формирует целочисленный массив размера N,

а затем находит сумму элементов между минимальным и максимальным элементами.

*Пример:*

[1 -2 **3** -4 **-6** 2 3] -> -4

**Состав**

Программа должна состоять из функций:

- int getSumMaxMin(int arr[],int N) - нахождение суммы в массиве arr длины N

- main().

Текст программы помещается в файлы: task6.h, task6.c, main6.c

**Задача №7**

Написать программу, которая печатает таблицу встречаемости

символов для введённой строки, отсортированную по убыванию частоты

**Состав**

Программа должна состоять из функций:

- main().

Текст программы помещается в файлы: main7.c