**Лабораторная №8**

«Дан текст. Слова в тексте разделены пробелами и знаками препинания. Получить n наиболее часто встречающихся слов и число их появлений. **Использовать рекурсию и функцию с переменным числом параметров**».

**Функция *main()***



**Функция *find()***

**1. НАЧАЛО**

2. Входные значения: массив строк *text*, содержащий введённый текст.

Промежуточные значения: символьная строка *temp* — буферная строка для копирования слов из *text*.

Выходные значения: массив строк *words*, содержащий все уникальные слова из *text*; массив чисел *array*, элементы которого содержат кол-во повторений слова в *text* (i-ый элемент *words* соответствует i-ому элементу *array*)

3. Выделение памяти для буферной строки *temp*.

4. В цикле, пока не будет достигнут конец массива, *text* поочерёдно перебираются все его элементы.

5. Если очередной элемент *text[i][j]* содержит знак препинания или пробел, то переходим к следующему элементу (*j++*).

6. Если очередной элемент *text[i][j]* содержит ‘\*0*’, то переходим к следующей строке (*i++; j = 0*).

7. Если элементы *text[i][j]* содержат буквы в верхнем регистре, то приводим их к нижнему регистру.

8. В temp c помощью функции ***copy()*** копируется очередное слово из текста с позиции *[i,j]*.

9. Далее оно сравнивается в уже записанными в массив *words* элементами с помощью функции ***compare()***, которая сравнивает посимвольно две строки. Если совпадения нет, то слово из *temp* записывается в *words*. Если совпадение есть, то увеличивается элемент массива *array*, соответствующий элементу массива *words*.

10. Переходим к концу слова, увеличивая счётчик *j* на длину слова из *temp*, получаемую с помощью функции *get\_length()*. Если конец массива не достигнут, то возвращаемся к пункту 3.

11. Выполняем сортировку массивов *words* и *array* методом Хоара, вызывая рекурсивную функцию ***hoor()***, которая делит массив на две части, повторно вызывая саму себя.

**12. КОНЕЦ**