**Лабораторная работа №5**

«Дано математическое выражение, в котором имеется несколько уровней вложенных скобок. необходимо удостоверится, что скобки расставлены правильно».

struct stack  
{

int pos; //позиция скобки в строке  
char bracket //тип скобки  
stack\* next; //указатель на следующий элемент

};

**Блок-схема main()**

****

**Функция check()**

**1. НАЧАЛО**

2. Входные значения: *str* — анализируемая строка, *\*st* — указатель на стек, *k* — целое число, содержащее кол-во ошибок.

Выходные значения: *arr[]* — массив, содержащий позиции ошибок в строке str

3. Выделение памяти для массива *arr[]* на кол-во символов в *str*

4. Анализ *str*. Если встречена открывающая скобка ‘(’, ‘{’ или ‘[’, то поместить её в стек *st* с помощью *push*().

5. Если встречена закрывающая скобка ‘)’, ‘}’ или ‘]’ и cтек *st* пуст, то занести позицию закрывающей скобки в *arr[]* и увеличить *k* на 1.

6. Если встречена закрывающая скобка ‘)’, ‘}’ или ‘]’ и последняя скобка в *st* того же типа то удалить открывающую скобку из *st* с помощью функции *pop*(). Если типы скобок не совпадают, то занести позицию закрывающей скобки в *arr[]* и увеличить *k* на 1.

7. Если не достигнут конец строки, то перейти к **пункту 4**, иначе — к **пункту** **8**.

8. Если в стеке остались открывающие скобки, то удалить их оттуда с помощью функции *pop*(), занося их позиции в *str* в *arr[]* и увеличивая *k*.

9. Перевыделение памяти для *arr*[] с учётом его настоящего размера *k*.

10. Сортировка массива *arr*[].

**11. КОНЕЦ**