19) EXPRESSION CONTAINS REDUNDANT BRACKETS OR NOT

METHOD:
LINK OF EXPLANATION: Redundant Parenthesis | C++ Placement Course | Lecture 25.4
Duplicate Brackets | Solution
This sum is done using stacks as working with parenthesis is involved.
In this sum, the method is, we push only the opening brackets and the operators into the stack.
For closing parenthesis,if we get the opening parenthesis at the top of the stack then we return true i.e the string contains the redundant parenthesis.
If for the closing parenthesis ,there is no opening parenthesis at the top then we pop the elements till we encounter the opening parenthesis.

CODE OF THE PROGRAM:

```
#include<iostream>
#include<stack>
using namespace std;

bool func(string &s){
   stack<char> st;
   for(int i=0;i<s.length();i++){
      if(s[i]=='(' || s[i]=='+'|| s[i]=='-'|| s[i]=='*'|| s[i]=='/'){
         st.push(s[i]);
      }
      else if(s[i]==')'){
         if(st.top()=='('){
            return true;
         }
         else{
            while(st.top()!='('){
               st.pop();
            }
            st.pop();
         }
      }
   }
   return false;
}

int main(){
   string s;
   cout<<"\n Enter the string:";
   cin>>s;
   cout<<"\n The string is redundant:"<<func(s);
   return 0;
}
```