

```
/*
```

In any language program mostly syntax error occurs due to unbalancing delimiter such as

(,},{,[]). Write C++ program using stack to check whether given expression is well parenthesized or not.

```
*/
```

```
#include <iostream>
```

```
using namespace std;
```

```
#define size 10
```

```
class stackexp {
```

```
    int top;
```

```
    char stk[size];
```

```
public:
```

```
    stackexp() {
```

```
        top = -1;
```

```
    }
```

```
    void push(char);
```

```
    char pop();
```

```
    int isfull();
```

```
    int isempty();
```

```
};
```

```
void stackexp::push(char x) {
```

```
    top = top + 1;
```

```
    stk[top] = x;
```

```
}
```

```
char stackexp::pop() {
```

```
    char s;
```

```
    s = stk[top];
```

```
    top = top - 1;
```

```
    return s;
```

```
}
```

```
int stackexp::isfull() {
```

```
    if (top == size)
```

```
        return 1;
```

```
    else
```

```
        return 0;
```

```
}
```

```
int stackexp::isempty() {
```

```
    if (top == -1)
```

```
        return 1;
```

```
    else
```

```
        return 0;
```

```
}
```

```
int main() {
```

```
    stackexp s1;
```

```

char exp[20], ch;

int i = 0;

cout << "\n\t!! Parenthesis Checker..!!!" << endl;

cout << "\nEnter the expression to check whether it is well-formed or not: ";

cin >> exp;

if ((exp[0] == '(' || exp[0] == '[' || exp[0] == '{')) {

    cout << "\nInvalid Expression.....\n";

    return 0;

} else {

    while (exp[i] != '\0') {

        ch = exp[i];

        switch (ch) {

            case '(': s1.push(ch); break;

            case '[': s1.push(ch); break;

            case '{': s1.push(ch); break;

            case ')': s1.pop(); break;

            case ']': s1.pop(); break;

            case '}': s1.pop(); break;

        }

        i = i + 1;

    }

}

if (s1.isEmpty()) {

    cout << "\nExpression is well parenthesized...\n";

} else {

```

```
        cout << "\nSorry!!! Invalid Expression or not well parenthesized....\n";  
    }  
    return 0;  
}
```