Experiment no. 2

Objective: Evalution of Paranthesis

Code:

```
#include <stdio.h>
#define MAX 20

struct stack
{
char stk[MAX];
int top;
}s;

void push(char item)
{
if (s.top == (MAX - 1))
printf ("Stack is Full\n");
else
{
s.top = s.top + 1;
s.stk[s.top] = item;
}}

void pop()
{
if (s.top == - 1)
```

```
{
printf ("Stack is Empty\n");
}
else
{
s.top = s.top - 1;
}}

int main()
{
char exp[MAX];
int i = 0;
s.top = -1;
printf("\nINPUT THE EXPRESSION : ");
scanf("%s", exp);
for(i = 0;i < strlen(exp);i++)
{
if(exp[i] == '(' || exp[i] == '[' || exp[i] == '{')
{
push(exp[i]);
continue;
}
else if(exp[i] == ')' || exp[i] == ']' || exp[i] == '}')
{
if(exp[i] == ')')
{
if(s.stk[s.top] == '(')
```

```
{
pop();
}
else
{
printf("\nUNBALANCED EXPRESSION\n");
break;
}}
if(exp[i] == ']')
{
if(s.stk[s.top] == '[')
{
pop();
}
else
{
printf("\nUNBALANCED EXPRESSION\n");
break;
}}
if(exp[i] == '}')
{
if(s.stk[s.top] == '{')
{
pop();
}
else
{
```

```
printf("\nUNBALANCED EXPRESSION\n");
break;
}
}
}
}
if(s.top == -1)
{
printf("\nBALANCED EXPRESSION\n");
}
}
```

Output:

```
INPUT THE EXPRESSION : [(])

UNBALANCED EXPRESSION

[Process completed - press Enter]
```

```
INPUT THE EXPRESSION : {[()]}

BALANCED EXPRESSION

[Process completed - press Enter]
```

Presented by: Gelle Hruthesh reddy (20BCB7031)