

## DSA\_College\validParentheses.cpp

```
1  #include<iostream>
2  #include<stack>
3  using namespace std;
4
5  struct node
6  {
7      int data ;
8      struct node* next ;
9  };
10 struct node* top =0 ;
11
12 bool isEmpty()
13 {
14     return top == nullptr ;
15 }
16
17 void push(char x)
18 {
19     struct node* newnode = (struct node*)malloc(sizeof(struct node)) ;
20     newnode->data = x;
21     newnode->next = top ;
22     top = newnode ;
23 }
24
25 void pop()
26 {
27     node* temp = top ;
28     top= top->next ;
29     free(temp) ;
30 }
31
32 bool isValid(string s)
33 {
34     for(int i=0;i<s.length();i++)
35     {
36         if(s[i] == '(' || s[i] == '[' || s[i] == '{')
37         {
38             push(s[i]) ;
39         }
40         else if(s[i] == ')' || s[i] == ']' || s[i] == '}')
41         {
42             if(isEmpty())
43             {
44                 return false ;
45             }
46             char topChar = top->data ;
47             if((s[i] == ')' && topChar != '(') || (s[i] == ']' && topChar != '[') || (s[i] ==
48             '{') && topChar != '{'))
49             {
50                 return false ;
51             }
52             pop() ;
53         }
54     }
55     return true ;
56 }
```

```
52     }
53 }
54 return isEmpty() ;
55 }
56
57 int main()
58 {
59     string s = "()){([]]}" ;
60     bool res = isValid(s) ;
61     if(res)
62     {
63         cout<<"valid parentheses"<<endl;
64     }
65     else{
66         cout<<"not valid parentheses"<<endl;
67     }
68     return 0;
69 }
```