DSA_College\validParentheses.cpp

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

struct node
{
    int data ;
    struct node* next ;
};
struct node* top =0 ;

bool isEmpty()
{
    return top == nullptr ;
}

void push(char x)
{
   struct node* newnode = (struct node*)malloc(sizeof(struct node)) ;
   newnode->data = x;
   newnode->next = top ;
   top = newnode ;
}

void pop()
{
    node* temp = top ;
    top= top->next ;
    free(temp) ;
}

bool isValid(string s)
{
    for(int i=0;i<s.length();i++)
    {
        if(s[i] == '(' || s[i] == '[' || s[i] == '{')
        {
            push(s[i]) ;
        }
        else if(s[i] == ')' || s[i] == ']' || s[i] == '}')
        {
            if(isEmpty())
            {
                return false ;
            }
            char topChar = top->data ;
            if((s[i] == ')' && topChar != '(') || (s[i] == ']' && topChar != '[') || (s[i] == '}' && topChar != '{'))
            {
                return false ;
            }
            pop() ;
        }
    }
    return isEmpty() ;
}

int main()
{
    string s = "()){([]})" ;
    bool res = isValid(s) ;
    if(res)
    {
        cout<<"valid parentheses"<<endl;
    }
    else{
         cout<<"not valid parentheses"<<endl;
    }
    return 0;
}
```