

```
1 #include <iostream>
2 #include <fstream>
3 #include <string>
4 #include <cstring>
5 using namespace std;
6 template <class t1>
7 struct st
8 {
9     t1 info;
10    st* next;
11 };
12
13 template <class t1>
14 class stack
15 {
16     st<t1> *top;
17 public:
18     stack();
19     void push(t1);
20     t1 pop();
21     bool isempty();
22     bool isfull();
23     void print();
24
25 };
26
27 template <class t1>
28 stack<t1>::stack()
29 {
30     top = NULL;
31 }
32
33 template<class t1>
34 void stack<t1>::push(t1 a)
35 {
36     st<t1>* temp;
37     temp = new st<t1>;
38     temp->info = a;
39     temp->next = top;
40     top = temp;
41 }
42
43 template<class t1>
44 t1 stack<t1>::pop()
45 {
46     t1 value;
47     st<t1>* temp;
48     value = top->info;
49     temp = top;
```

```
50     top = top->next;
51     temp->next = NULL;
52     delete temp;
53     return value;
54 }
55
56 template<class t1>
57 bool stack<t1>::isempty()
58 {
59     if (top == NULL)
60     {
61         return true;
62     }
63     else
64         return false;
65 }
66
67 template<class t1>
68 bool stack<t1>::isfull()
69 {
70     st<t1>* ptr;
71     ptr = new st<t1>;
72     if (ptr == NULL)
73     {
74         return true;
75     }
76     else
77     {
78         delete ptr;
79         return false;
80     }
81 }
82
83 template<class t1>
84 void stack<t1>::print()
85 {
86     st<t1>* temp;
87     temp = top;
88     while (temp != NULL)
89     {
90         cout << temp->info;
91         temp = temp->next;
92     }
93 }
```