

```

1  #include <iostream>
2  using namespace std ;
3
4  struct elemenet
5  {
6      int          data ;
7      struct elemenet * next ;
8  };
9
10 typedef struct elemenet Element ;
11 typedef Element * Stack ;
12
13
14 Stack create_stack()
15 {
16     return NULL ;
17 }
18
19 bool empty_stack(Stack stk)
20 {
21     if(stk == NULL)
22         return true ;
23     else
24         return false ;
25 }
26
27 int depth_stack(Stack stk)
28 {
29     Stack ptr = stk ;
30     int count = 0 ;
31
32     while(ptr != NULL)
33     {
34         count++ ;
35         ptr = ptr->next ;
36     }
37
38     return count ;
39 }
40
41 Stack push(Stack stk, int val)
42 {
43     Stack ptr = new Element ;
44     ptr->data = val ;
45
46     ptr->next = stk ;
47     stk = ptr ;
48
49     return stk ;
50 }
51
52 Stack pop(Stack stk, int &val)
53 {
54     if(!empty_stack(stk))
55     {
56         Stack ptr = stk ;
57
58         val = stk->data ;
59         stk = stk->next ;
60         delete ptr ;
61
62         return stk ;
63     }
64     else
65     {
66         cout << "Error, the stack is already empty." << endl ;
67         return NULL ;
68     }
69 }
70
71 int top_stack(Stack stk)
72 {
73     if(empty_stack(stk))
74     {
75         cout << "Error, Stack is empty." << endl ;
76         return 0 ;
77     }
78

```

```

79     return stk->data ;
80 }
81
82 Stack replace_top_stack(Stack stk, int val)
83 {
84     if(empty_stack(stk))
85     {
86         cout << "Error, Stack is empty." << endl ;
87         return NULL ;
88     }
89
90     stk->data = val ;
91
92     return stk ;
93 }
94
95 void display_stack(Stack stk)
96 {
97     Stack ptr = stk ;
98
99     if(empty_stack(stk))
100    {
101        cout << "Stack is empty." << endl ;
102        return ;
103    }
104
105    while(ptr != NULL)
106    {
107        cout << ptr->data << " " ;
108        ptr = ptr->next ;
109    }
110    cout << endl ;
111
112    return ;
113 }
114
115 Stack reverse_stack(Stack stk)
116 {
117     Stack tmp = create_stack() ;
118
119     int val ;
120
121     while(!empty_stack(stk))
122     {
123         stk = pop(stk, val) ;
124         tmp = push(tmp, val) ;
125     }
126
127     return tmp ;
128 }
129
130 int main()
131 {
132     Stack stk = create_stack() ;
133
134     stk = push(stk, 1) ;
135     stk = push(stk, 2) ;
136     stk = push(stk, 3) ;
137
138     display_stack(stk) ;
139
140     stk = reverse_stack(stk) ;
141
142     display_stack(stk) ;
143
144     return 0 ;
145 }

```