

**stack\_out.c**

```
1 #include<stdio.h>
2 #define maxsize 5
3 int stack [maxsize];
4 int top=-1;
5
6 void push (int item){
7     if (top>=maxsize-1){
8         printf("The stack is full\n");
9     }
10
11    else
12    {
13        top++;
14        stack[top]=item;
15    }
16}
17
18 int pop()
19 {
20     if (top<0)
21     {
22         printf("the stack is empty\n");
23         return 0;
24     }
25
26    else
27    {
28        int item=stack[top];
29        top=top-1;
30        return item;
31    }
32}
33
34 void display()
35 {
36     if (top<0)
37     {
38         printf("The stack is empty\n");
39     }
40
41
42    else{
43        for (int i=0; i<=top; i++)
44        {
45            printf("The elements are : %d \n",stack[i]);
46        }
47    }
48 }
```

```
49
50 int main(){
51     int choice, item;
52
53     while(1)
54     {
55         printf("\n");
56         printf("1. Push\n2. Pop\n3. Display\n4. Exit\n");
57
58         printf("Enter choice : ");
59         scanf("%d",&choice);
60
61         switch (choice)
62         {
63             case 1: printf("Choice selected : \"PUSH\" \n");
64                 printf("Enter an item to insert : ");
65                 scanf("%d",&item);
66                 push(item);
67                 break;
68             case 2: printf("Choice selected : \"POP\" \n");
69                 printf("Enter item to be removed : %d\n",pop());
70                 break;
71             case 3: printf("Choice selected : \"DISPLAY\" \n");
72                 display();
73                 break;
74             case 4: return 0;
75             default: printf("Invalid Choice\n");
76         }
77     }
78 }
```