

```

1  #include<stdio.h>
2  #include<stdlib.h>
3
4  struct stack
5  {
6      int top;
7      int capacity;
8      int *array;
9  };
10
11
12
13  struct stack *createStack()
14  {
15      struct stack *S=malloc(sizeof( struct stack));
16      S->top=-1;
17      S->capacity=1;
18      S->array=malloc(S->capacity*sizeof(int));
19
20      return S;
21  }
22
23  int isEmpty(struct stack *S)
24  {
25      if(S->top== -1)
26          return 1;
27      else
28          return 0;
29  }
30
31  int isFull(struct stack *S)
32  {
33      if(S->top==S->capacity-1)
34          return 1;
35      else
36          return 0;
37  }
38
39  void push(struct stack *S,int k)
40  {
41      if(isFull(S))
42
43          printf("OverFlow");
44
45      else
46
47
48
49          S->array[++S->top]=k;
50
51
52  }
53
54  void pop(struct stack *S)
55  {
56      int data;
57      if(isEmpty(S))
58          printf("UnderFlow");
59      else
60      {
61          data=S->array[S->top--];
62      }
63  }
64
65  void main()
66  {

```

```
67
68  struct stack *S=createStack();
69  push(S,3);
70  push(S,3);
71  pop(S);
72  printf("%d",isEmpty(S));
73  }
```