

```
1  //implement stack
2  class Stack
3  {
4      private int size, index, arr[];
5      Stack(int n)
6      {
7          size = n;
8          arr = new int[n];
9          index = -1;
10     }
11     protected Boolean isEmpty()
12     {
13         if(index == -1)
14             return true;
15         return false;
16     };
17     protected Boolean isFull()
18     {
19         if(index==size-1)
20             return true;
21         return false;
22     }
23     int Top()
24     {
25         if(!isEmpty())
26         {
27             return arr[index];
28         }
29         System.out.println("Stack underflow");
30         return -1;
31     }
32     void Push(int x)
33     {
34         if(isFull())
35         {
36             System.out.println("stack overflow");
37             return;
38         }
39         index++;
40         arr[index] = x;
41         System.out.println(x + " is pushed into stack");
42     }
43     int Pop()
44     {
45         if(isEmpty())
46         {
47             System.out.println("Stack underflow");
48             return -1;
49         }
50         int p = arr[index];
```

```
51         index-- ;
52         return p;
53     }
54     protected void display()
55     {
56         for(int i=0; i<=index; i++)
57         {
58             System.out.print(arr[i]+" ", );
59         }
60     }
61
62 }
63
64 class Question2
65 {
66     public static void main(String arg[])
67     {
68         Stack s = new Stack(5);
69         s.Pop();
70         s.Push(4);
71         s.Push(5);
72         s.Push(7);
73         s.Push(9);
74         System.out.println(s.isEmpty());
75         s.Push(1);
76         System.out.println(s.isFull());
77         s.Push(2);
78         System.out.println(s.isFull());
79         System.out.println(s.Pop());
80         s.display();
81     }
82 }
83 //output
84 // PS C:\Users\Dell\Desktop\JAVA\Assignments\Assignment-3> java Question2
85 // Stack underflow
86 // 4 is pushed into stack
87 // 5 is pushed into stack
88 // 7 is pushed into stack
89 // 9 is pushed into stack
90 // false
91 // 1 is pushed into stack
92 // true
93 // stack overflow
94 // true
95 // 1
96 // 4, 5, 7, 9,
97
98
99
```