```
1
2   // implementing the Stack ADT using array
3
4   class Stack{
5
6       private int array[];
7       private int maxSize; // useful to decide if resizing (doubling the array size) is
        needed
8       private int topOfStack; // same as endOfArray
9
10      public Stack(int size){
11              array = new int[size];
12              maxSize = size;
13              topOfStack = -1;
14      }
15
16      public boolean isEmpty(){
17
18              if (topOfStack == -1)
19                  return true;
20
21              return false;
22      }
23
24      public  void resize(int s){
25
26              int tempArray[] = array;
27
28              array = new int[s];
29
30              for (int index = 0; index < Math.min(s, topOfStack+1); index++){
31                  array[index] = tempArray[index];
32              }
33
34              maxSize = s;
35
36      }
37
38
39      public  void push(int data){    // same as insert 'at the end'
40
41              if (topOfStack == maxSize-1)
42                  resize(2*maxSize);
43
44              array[++topOfStack] = data;
45
46      }
47
48
49      public  int peek(){
50
51              if (topOfStack >= 0)
52                  return array[topOfStack];
53              else
54                  return -1000000;// an invalid value indicating
55                                  // stack is empty
56
57      }
58
59
60      public  int pop(){
61
62              if (topOfStack >= 0){
63                  return array[topOfStack--];
```

```
                // the topOfStack is decreased by one
            }
            else
                return -1000000; // an invalid value indicating
                                 // stack is empty
    }



}


class DynamicArrayBasedStack{

    public static void main(String[] args){

    Stack stack = new Stack(1);

    stack.push(10);
    stack.push(23);
    stack.push(100);
    stack.push(45);

    System.out.println("pop: " + stack.pop() );
    System.out.println("peek at top: " + stack.peek() );

    stack.push(85);
    stack.push(12);

    while (!stack.isEmpty())
        System.out.println("pop: " + stack.pop() );


   }

}
```

```
pop: 45
peek at top: 100
pop: 12
pop: 85
pop: 100
pop: 23
pop: 10
```