

Code : [LP\\_Code1.java](#)

Output :

```
Elements of queue [0, 1, 2, 3, 4, 5, 6, 7, 8, 9]
removed element-0
[1, 2, 3, 4, 5, 6, 7, 8, 9]
head of queue-1
Size of queue-9
```

**Q2. Write a program to implement the functionalities of a stack using the methods of queue only.**

**Solution :**

Code : [LP\\_Code2.java](#)

Output :

```
current size: 6
The peek element is :
6
The peek element is :
5
The peek element is :
4
current size: 4
```

**Approach :**

- The idea behind this approach is to make one queue and push the first element in it.
- After the first element, we push the next element and then push the first element again and finally pop the first element.
- So, according to the FIFO rule of the queue, the second element that was inserted will be at the front and then the first element as it was pushed again later and its first copy popped out.
- So, this acts as a Stack and we do this at every step i.e. from the initial element to the second last element, and the last element will be the one that we are inserting and since we will be pushing the initial elements after pushing the last element, our last element becomes the first element.

**Time Complexity:**

- Push operation:  $O(N)$
- Pop operation:  $O(1)$

**Space complexity:**  $O(N)$  since 1 queue is used.

**Q3. Write a program to implement the functionalities of a queue using the methods of stack only.**

**Solution :**

Code : [LP\\_Code3.java](#)