## Implement the Min Priority Queue using Heap data structure
## Code : Remove Min

Send Feedback

Implement the function RemoveMin for the min priority queue class.

For a minimum priority queue, write the function for removing the minimum element present.

Remove and return the minimum element.

**Note : main function is given for your reference which we are using internally to test the code.**

The main source code is –

```
import java.io.BufferedReader;
import java.io.IOException;
import java.io.InputStreamReader;
import java.util.StringTokenizer;

public class Main {

	static BufferedReader br = new BufferedReader(new
InputStreamReader(System.in));;
	static StringTokenizer st;

	public static void main(String[] args) throws
NumberFormatException, IOException {
		st = new StringTokenizer(br.readLine());
		PQ pq = new PQ();
		int choice = Integer.parseInt(st.nextToken());
		while(choice != -1) {
			switch(choice) {
				case 1 : // insert
					int element =
Integer.parseInt(st.nextToken());
					pq.insert(element);
					break;
				case 2 : // getMin
				try {
					System.out.println(pq.getMin());
				} catch (PriorityQueueException e) {
					System.out.println(Integer.MIN_VALUE);
					return;
				}
					break;
				case 3 : // removeMin
				try {
					System.out.println(pq.removeMin());
				} catch (PriorityQueueException e) {
					System.out.println(Integer.MIN_VALUE);
					return;
				}
					break;
				case 4 : // size
					System.out.println(pq.size());
					break;
```

```
                        case 5 : // isEmpty
                              System.out.println(pq.isEmpty());
                        default :
                                    return;
                  }
                  choice = Integer.parseInt(st.nextToken());
            }
      }

}
```