

```

import java.util.*;
public class ShowMeBug {
    public static void main(String[] args) {
        //1.
        System.out.println("1.Hello");

        //2.
        int[] nums = new int[2];
        int[] testCase = {12,12,21};
        Arrays.sort(nums);
        System.out.println("2.nums.length:" + nums.length);
        int[][] arr = new int[2][2];
        System.out.println("2.arr.length:" + arr.length);

        //3.
        String str = "sss";
        System.out.println("3.str.length:" + str.length());
        str = str.replace('s','a');
        System.out.println("3.str.charAt():" + str.charAt(0));
        System.out.println("3.str.equals():" + str.equals("aaa"));

        //4.
        Stack stack = new Stack();
        stack.push(12);stack.push(13);
        int temp = (int)stack.pop();
        System.out.println("4.stack.peek():" + stack.peek());
        System.out.println("4.stack.size():" + stack.size());

        //5.
        Vector v = new Vector(4);
        v.add("Test0");
        v.add("Test1");
        v.remove(0);
        int size = v.size();
        System.out.println("4.v.get(0):" + v.get(0));
        List<int[]> vec = new ArrayList<int[]>();
        vec.toArray(new int[vec.size()][]);

        //6.
        Map<Character,Integer> map = new HashMap<Character,Integer>();
        Map<Integer, Boolean> hash = new HashMap<Integer,Boolean>();
        hash.put(12,false);
        hash.containsKey("");
        for (Integer key : hash.keySet()) {
            System.out.println("6. Key = " + key);
            System.out.println("6 .value = " + hash.get(key));
        }

        //7.List
    }
}

```

```
List list = new ArrayList<>();
list.add("apple");
list.add("apple");
System.out.println("7 .list = " + list.size());
```

```
//8. Queue
Queue queue = new LinkedList<>();
queue.add(1);
queue.add(2);
queue.add(3);
queue.isEmpty();
queue.poll();
System.out.println("8 .queue " + queue.peek());
```

```
//9. Deque
Deque mDeque = new ArrayDeque();
//FIFO
mDeque.add(1);
mDeque.add(2);
mDeque.add(3);
mDeque.poll();
mDeque.add(3);
mDeque.isEmpty();
//Stack
mDeque.push(111);
System.out.println("9 .Deque " + mDeque.pop());
```

```
//10. 最小
PriorityQueue Pqueue = new PriorityQueue();
Pqueue.offer(1);
Pqueue.offer(2);
Pqueue.offer(3);
System.out.println("10 .PriorityQueue " + Pqueue.poll());
```

```
}
}
```