**Part A**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Operation No.** | **Operation** | **Operation Result** | **No. of Items on the Queue** | **Queue** |
| 1 | enqueue(‘Ireland’) | ‘Ireland’ | 1 | ‘Ireland’ |
| 2 | dequeue() |  | 0 |  |
| 3 | enqueue(‘England’) | ‘England’ | 1 | ‘England’ |
| 4 | dequeue() |  | 0 |  |
| 5 | enqueue(‘Wales’) | ‘Wales’ | 1 | ‘Wales’ |
| 6 | dequeue() |  | 0 |  |
| 7 | enqueue(‘Scotland’) | ‘Scotland’ | 1 | ‘Scotland’ |
| 8 | dequeue() |  | 0 |  |
| 9 | enqueue(‘France’) | ‘France’ | 1 | ‘France’ |
| 10 | enqueue(‘Germany’) | ‘Germany’ | 2 | ‘Germany’,‘France’ |

**Q3.**

**Q4.** Algorithm insertLast(o):

*Input: an object o*

*Output: N/A*

node ← New Node(o)

node.prev ← rear

**if** rear = null **then**

front ← node

**else**

rear.next ← node

rear ← node

size ← size+1

**Q5.** Algorithm removeLast():

*Input: N/A*

*Output: N/A*

**if** size > 0 **then**

node ← node.prev

**if** node = null **then**

front ← null

rear ← null

**else**

node.next ← null

rear ← node

size ← size-1

Algorithm removeFirst(o):

*Input: N/A*

*Output: N/A*

if size > 0 then

node ← first

node.next ← node

if node = null then

front ← null

rear ← null

else

node.prev ← null

first ← node

size ← size-1

**Q6.**

Algorithm isEmpty():

*Input: N/A*

*Output: Boolean true/false*

**if** size == 0 **then**

**return** true

**else**

**return** false

Algorithm size():

*Input: N/A*

*Output: Integer of Size*

**return** size

Algorithm front():

*Input: N/A*

*Output: Front Node*

**return** (Node) front

Algorithm rear():

*Input: N/A*

*Output: Rear Node*

**return** (Node) rear

**Q7.**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Operation No.** | **Operation** | **Operation Result** | **No. of Items on the Queue** | **Queue** |
| 1 | insertFirst(‘Ireland’) | ‘Ireland’ | 1 | ‘Ireland’ |
| 2 | removeLast() |  | 0 |  |
| 3 | insertLast(‘England’) | ‘England’ | 1 | ‘England’ |
| 4 | removeFirst() |  | 0 |  |
| 5 | insertLast(‘Wales’) | ‘Wales’ | 1 | ‘Wales’ |
| 6 | insertFirst(‘Scotland’) | ‘Scotland’ | 2 | ‘Wales’,‘Scotland’ |
| 7 | insertLast(‘France’) | ‘France’ | 3 | ‘France’,‘Wales’,‘Scotland’ |
| 8 | removeFirst() |  | 2 | ‘France’,‘Wales’ |
| 9 | removeLast() |  | 1 | ‘Wales’ |
| 10 | insertLast(‘Germany’) |  | 2 | ‘Germany’,‘Wales’ |