**Name:** Michael Mullarkey (16204743)

**Part A**

**3)**

Array Based

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Operation | F | R | 0 | 1 | 2 | 3 | 4 | 5 |
| Initial State | 0 | 0 |  |  |  |  |  |  |
| enqueue(‘Ireland’) | 0 | 1 | Ireland |  |  |  |  |  |
| dequeue() | 1 | 1 |  |  |  |  |  |  |
| enqueue(‘England’) | 1 | 2 |  | England |  |  |  |  |
| dequeue() | 2 | 2 |  |  |  |  |  |  |
| enqueue(‘Wales’) | 2 | 3 |  |  | Wales |  |  |  |
| dequeue() | 3 | 3 |  |  |  |  |  |  |
| enqueue(‘Scotland’) | 3 | 4 |  |  |  | Scotland |  |  |
| dequeue() | 4 | 4 |  |  |  |  |  |  |
| enqueue(‘France’) | 4 | 5 |  |  |  |  | France |  |
| enqueue(‘Germany’) | 4 | 6 |  |  |  |  | France | Germany |

Link Based

|  |  |  |
| --- | --- | --- |
| Operation | 0 | 1 |
| enqueue(‘Ireland’) | [Ireland]-> |  |
| dequeue() |  |  |
| enqueue(‘England’) | [England]-> |  |
| dequeue() |  |  |
| enqueue(‘Wales’) | [Wales]-> |  |
| dequeue() |  |  |
| enqueue(‘Scotland’) | [Scotland]-> |  |
| dequeue() |  |  |
| enqueue(‘France’) | [France]-> |  |
| enqueue(‘Germany’) | [France]-> | [Germany]-> |

**Countries Held:** France, Germany

**Countries dequeued:** Ireland, England, Wales, Scotland

**4)**

*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*

Dry run 1 – empty deque

|  |  |  |  |
| --- | --- | --- | --- |
| Operation | 0 | 1 | 2 |
| addLast("Ireland") | [Ireland]-> |  |  |
| addLast("Spain") | [Ireland]-> | [Spain]-> |  |
| addLast("Iceland") | [Ireland]-> | [Spain]-> | [Iceland]-> |

Dry run 2

Queue = [Ireland]-> [Spain]-> [Iceland]->

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Operation | 0 | 1 | 2 | 3 | 4 |
| addLast("Ireland") | [Ireland]-> |  |  |  |  |
| addLast("Spain") | [Ireland]-> | [Spain]-> |  |  |  |
| addLast("Iceland") | [Ireland]-> | [Spain]-> | [Iceland]-> |  |  |
| addLast("Brazil") | [Ireland]-> | [Spain]-> | [Iceland]-> | [Brazil]-> |  |
| addLast("Hungary") | [Ireland]-> | [Spain]-> | [Iceland]-> | [Brazil]-> | [Hungary]-> |

**5)**

*Algorithm removeLast()*

*Input: none*

*Output: the node removed*

*node <- rear*

*rear <- rear.prev*

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

*front <- null*

***else***

*rear.next <- null*

*size <- size – 1*

***return*** *node.element*

Dry run 1 – 1 item

|  |  |
| --- | --- |
| Operation | 0 |
| addLast("Ireland") | [Ireland]-> |
| removeLast() |  |

Dry run 2 – more than 1 item

Queue = [Ireland]-> [Spain]-> [Iceland]->

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Operation | 0 | 1 | 2 | 3 |
| addLast("Brazil") | [Ireland]-> | [Spain]-> | [Iceland]-> | [Brazil]-> |
| removeLast() | [Ireland]-> | [Spain]-> | [Iceland]-> |  |
| removeLast() | [Ireland]-> | [Spain]-> |  |  |
| removeLast() | [Ireland]-> |  |  |  |

*Algorithm removeFirst()*

*Input: none*

*Output: the node removed*

*node <- front*

*front <- front.next*

***if*** *front = null* ***then***

*rear <- null*

***else***

*node.next <- null*

*size <- size – 1*

***return*** *node.element*

Dry run 1 – 1 item

|  |  |
| --- | --- |
| Operation | 0 |
| addLast("Ireland") | [Ireland]-> |
| removeFirst() |  |

Dry run 2 – more than 1 item

Queue = [Ireland]-> [Spain]-> [Iceland]->

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Operation | 0 | 1 | 2 | 3 |
| addLast("Brazil") | [Ireland]-> | [Spain]-> | [Iceland]-> | [Brazil]-> |
| removeFirst() | [Spain]-> | [Iceland]-> | [Brazil]-> |  |
| removeFirst() | [Iceland]-> | [Brazil]-> |  |  |
| removeFirst() | [Brazil]-> |  |  |  |

**6)**

*Algorithm isEmpty()*

*Input: none*

*Output: true if the stack is empty, false otherwise*

***return*** *size == 0*

*Algorithm size()*

*Input: none*

*Output: count the size of the queue*

***return*** *size*

*Algorithm front()*

*Input: none*

*Output: top object*

***return*** *front.element*

*Algorithm rear()*

*Input: none*

*Output: bottom object*

***return*** *rear.element*

**7)**

|  |  |  |  |
| --- | --- | --- | --- |
| Operation | 0 | 1 | 2 |
| insertFirst(‘Ireland’) | [Ireland]-> |  |  |
| removeLast() |  |  |  |
| insertLast(‘England’) | [England]-> |  |  |
| removeFirst() |  |  |  |
| insertLast(‘Wales’) | [Wales]-> |  |  |
| insertFirst(‘Scotland’) | [Scotland]-> | [Wales]-> |  |
| insertLast(‘France’) | [Scotland]-> | [Wales]-> | [France]-> |
| removeFirst() | [Wales]-> | [France]-> |  |
| removeLast() | [Wales]-> |  |  |
| insertLast(‘Germany’) | [Wales]-> | [Germany]-> |  |

**Countries Held:** Wales, Germany

**Countries dequeued:** Ireland, England, Scotland, France