GIT Department of Computer Engineering

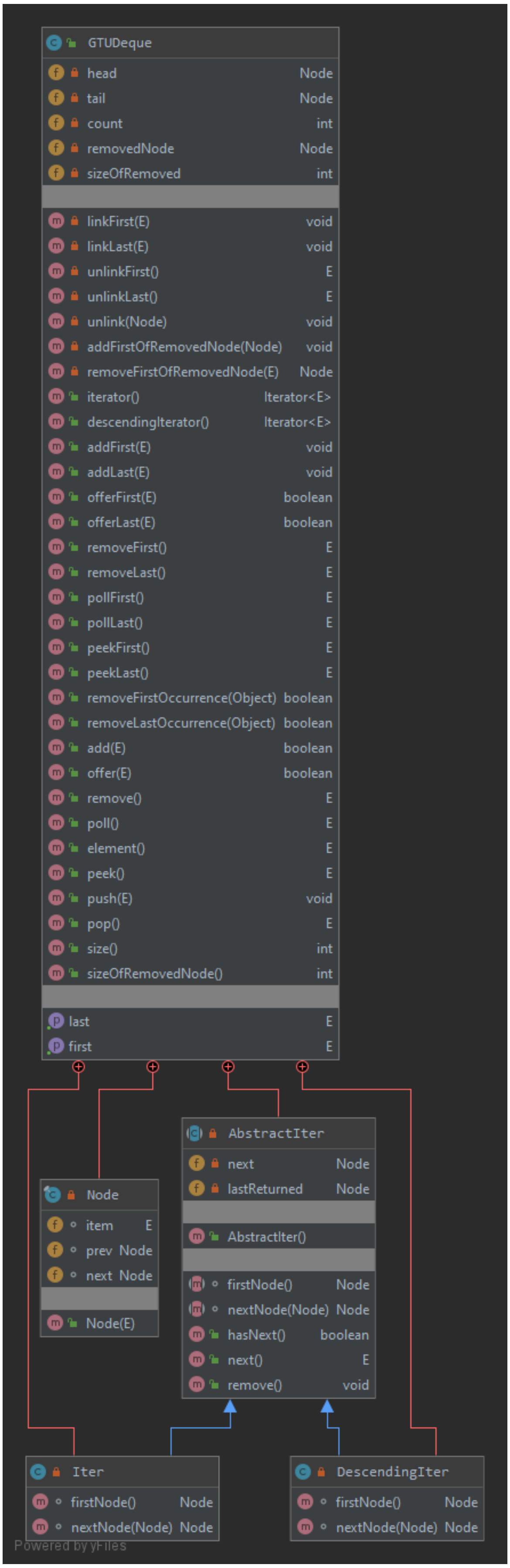
CSE 222/505 – Spring 2020

Homework #04 Part 2 Report

Abdullah ÇELİK

171044002

Class Diagram



Problem Solution Approach

Firstly, our requested deque class extends the AbstractCollection abstract class and implements the Deque interface. When I examined the Deque api, I understood that many methods are actually enough to call each other. For this reason, I firstly determined the methods I should write. These are offerFirst, offerLast, pollFirst, pollLast, peekFirst, peekLast. Before I wrote these methods, I knew that I had to carefully point the next and previous of our nodes. For this reason, I wrote functions in these processes. Then all I have to do is call the methods the right methods. To write the Iterator and descendingIterator methods, I had to write the iterator and descending iterator classes. If I wrote the methods of the two classes separately, the code would be duplicate. To avoid this, I wrote an abstract iterator class and it had two abstract methods in this class. These were the firstNode and nextNode method. For Iterator and descending iterator classes, I override these two methods correctly for these two classes to work properly.

Test Cases

Testing Methods of GTUDeque					
Test ID	**Scenerio**	**Test Data**	**Expected Results**	**Actual Results**	**Pass/Fail**
D01	int size() method called when deque is empty and has some elements	Size : 0 Size : 10	Successfully returned right size	As expected	Pass
D02	Iterator<E> iterator() method called when deque is empty and has some elements	Size : 0 Size : 5	Successfully iterator created	As expected	Pass
D03	Iterator<E> descendingIterator() method called when deque is empty and has some elements	Size : 0 Size : 5	Successfully descending iterator created	As expected	Pass
D04	boolean add(E e) method called when deque is empty and has some elements	Size : 0 e : 'X' e : 'Y' e : 'Z' Size : 3 e : 'K' e : 'L' e : 'M'	Successfully added and returned	As expected	Pass

D05	void addFirst(E e) method called when deque is empty and has some elements	Size : 0 e : 'X' e : 'Y' e : 'Z' Size : 3 e : 'K' e : 'L' e : 'M'	Successfully added first	As expected	Pass
D06	void addLast(E e) method called when deque is empty and has some elements	Size : 0 e : 'X' e : 'Y' e : 'Z' Size : 3 e : 'K' e : 'L' e : 'M'	Successfully added last	As expected	Pass
D07	boolean offer(E e) method called when deque is empty and has some elements	Size : 0 e : 'X' e : 'Y' e : 'Z' Size : 3 e : 'K' e : 'L' e : 'M'	Successfully added last and returned	As expected	Pass
D08	boolean offerFirst(E e) method called when deque is empty and has some elements	Size : 0 e : 'X' e : 'Y' e : 'Z' Size : 3 e : 'K' e : 'L' e : 'M'	Successfully added first and returned	As expected	Pass
D09	boolean offerLast(E e) method called when deque is empty and has some elements	Size : 0 e : 'X' e : 'Y' e : 'Z' Size : 3 e : 'K' e : 'L' e : 'M'	Successfully added last and returned	As expected	Pass

D10	E element() method called when deque has some elements	Size : 5	Successfully returned right element	As expected	Pass
D11	E element() method called when deque is empty	Size : 0	Expected NoSuchElement Exception	As expected	Pass
D12	E getFirst() method called when deque has some elements	Size : 5	Successfully returned first element	As expected	Pass
D13	E getFirst() method called when deque is empty	Size : 0	Expected NoSuchElement Exception	As expected	Pass
D14	E getLast() method called when deque has some elements	Size : 5	Successfully returned last element	As expected	Pass
D15	E getLast() method called when deque is empty	Size : 0	Expected NoSuchElement Exception	As expected	Pass
D16	E peek() method called when deque is empty and has some elements	Size : 0 Size : 5	Successfully returned first element if it is available otherwise returned null	As expected	Pass
D17	E peekFirst() method called when deque is empty and has some elements	Size : 0 Size : 5	Successfully returned first element if it is available otherwise returned null	As expected	Pass
D18	E peekLast() method called when deque is empty and has some elements	Size : 0 Size : 5	Successfully returned last element if it is available otherwise returned null	As expected	Pass
D19	E remove() method called when deque has some elements	Size : 5	Successfully removed first element and returned it	As expected	Pass
D20	E remove() method called when deque is empty	Size : 0	Expected NoSuchElement Exception	As expected	Pass
D21	E removeFirst() method called when deque has some elements	Size : 5	Successfully removed first element and returned it	As expected	Pass

D22	E removeFirst() method called when deque is empty	Size : 0	Expected NoSuchElement Exception	As expected	Pass
D23	E removeLast() method called when deque has some elements	Size : 5	Successfully removed last element and returned it	As expected	Pass
D24	E removeLast() method called when deque is empty	Size : 0	Expected NoSuchElement Exception	As expected	Pass
D25	E poll() method called when deque is empty and has some elements	Size : 0 Size : 5	Successfully removed first element and returned it if it is avaiable	As expected	Pass
D26	E pollFirst() method called when deque is empty and has some elements	Size : 0 Size : 5	Successfully removed first element and returned it if it is avaiable	As expected	Pass
D27	E pollLast() method called when deque is empty and has some elements	Size : 0 Size : 5	Successfully removed last element and returned it if it is avaiable	As expected	Pass
D28	boolean remove(Object o) method called when deque is empy and has some elements	Size : 0 o : 'X' Size : 5 o : 'B' o : 'X' o : 'D' o : 'B'	Successfully removed first occurence if it is available and returned	As expected	Pass
D29	boolean removeFirstOccurrence(Object o) method called when deque is empty and has some elements	Size : 0 o : 'X' Size : 5 o : 'D' o : 'X' o : 'A' o : 'B' o : 'A' o : 'B' o : 'B'	Successfully removed first occurence if it is available and returned	As expected	Pass

D30	boolean removeLastOccurrence(Object o) method called when deque is empty and has some elements	Size : 0 o : 'X' Size : 5 o : 'D' o : 'X' o : 'A' o : 'B' o : 'A' o : 'B' o : 'B'	Successfully removed last occurence if it is available and returned	As expected	Pass
D31	boolean contains(Object o) method called when deque is empty and has some elements	Size : 0 o : 'X' Size : 5 o : 'C'	Successfully returned true or false	As expected	Pass
D32	E pop() method called when deque has some elements	Size : 5	Successfully returned right elements and removed it	As expected	Pass
D33	E pop() method called when deque is empty	Size : 0	Expected NoSuchElement Exception	As expected	Pass
D34	void push(E e) method called when deque is empty	Size : 0 e : 'X' e : 'Y' e : 'Z'	Successfully added to first elements	As expected	Pass

Testing Methods of Iterator					
Test ID	**Scenerio**	**Test Data**	**Expected Results**	**Actual Results**	**Pass/Fail**
Iteator<E> iterator() method of GTUDeque called firstly in the following test					
I01	boolean hasNext() method called when deque is empty and has some elements	Size : 0 Size : 5	Successfully returned true or false	As expected	Pass
I02	E next() method called when deque has some elements	Size : 5	Successfully returned right element	As expected	Pass
I03	E next() method called when deque is empty	Size : 0	Expected NoSuchElement Exception	As expected	Pass

I04	void remove() method called after E next() method when deque has some elements	Size : 5	Successfully removed right element	As expected	Pass
I05	void remove() method called before E next() method when deque has some elements	Size : 5	Exptected IllegalStateException	As expected	Pass
Iteator<E> descendingIterator() method of GTUDeque called firstly in the following test					
DI01	boolean hasNext() method called when deque is empty and has some elements	Size : 0 Size : 5	Successfully returned true or false	As expected	Pass
DI02	E next() method called when deque has some elements	Size : 5	Successfully returned right element	As expected	Pass
DI03	E next() method called when deque is empty	Size : 0	Expected NoSuchElement Exception	As expected	Pass
DI04	void remove() method called after E next() method when deque has some elements	Size : 5	Successfully removed right element	As expected	Pass
DI05	void remove() method called before E next() method when deque has some elements	Size : 5	Exptected IllegalStateException	As expected	Pass

Testing Iterator Methods' Continuity

Test ID	Scenerio	Test Data	Expected Results	Actual Results	Pass/Fail
M01	Firstly Iterator<E> iterator() method called then while boolean hasNext() returned true, E next() method called	Size : 10	Successfully next method returned elements first to end	As expected	Pass

M02	Firstly Iterator<E> descendingIterator() method called then while boolean hasNext() returned true, E next() method called	Size : 10	Successfully next method returned elements end to first	As expected	Pass
M03	Firstly Iterator<E> iterator() method called then while boolean hasNext() returned true, E next() and E remove() methods called	Size : 10	Successfully removed method remove elements first to end	As expected	Pass
M04	Firstly Iterator<E> descendingIterator() method called then while boolean hasNext() returned true, E next() and E remove() methods called	Size : 10	Successfully removed method remove elements end to first	As expected	Pass

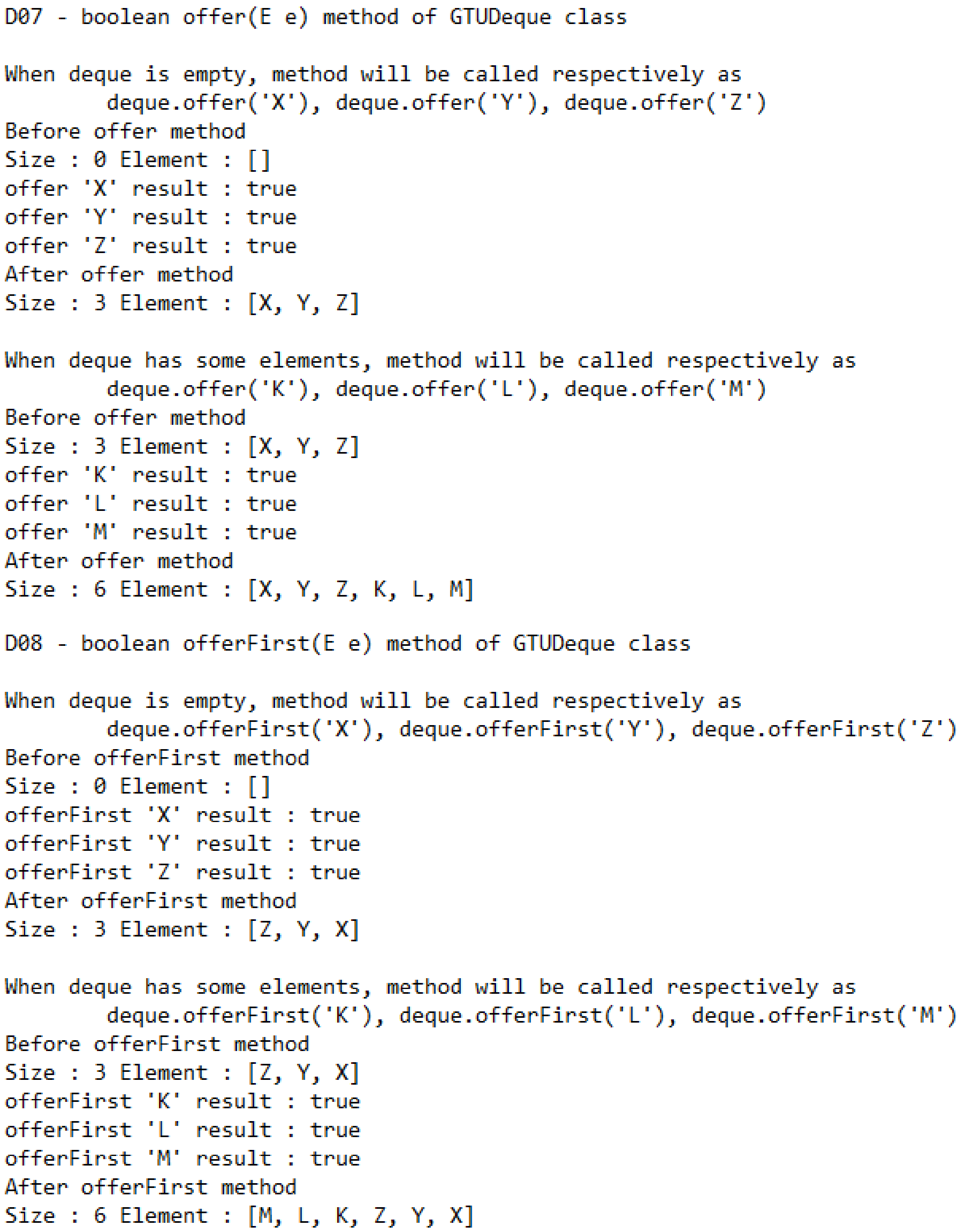
Testing Size of Deque and Size of Removed Node

Test ID	Scenerio	Test Data	Expected Results	Actual Results	Pass/Fail
S01	Respectively boolean offer(E e) boolean offerFirst(E e) E removeLast() E remove() boolean add(E e) void addFirst(E e)	Size of Deque : 0 Size of RemovedNode : 0	Successfully if removed node size is greater than 0, the nodes are taken from there and added. While the nodes are removed, the nodes are added to the removed node. Otherwise node was created	As expected	Pass

Running and Results

```
D01 - int size() method of GTUDeque class

When deque is empty
Size : 0 Element : []

When deque has some elements
Size : 10 Element : [A, B, C, D, E, F, G, H, I, J]

D02 - Iterator<E> iterator() method of GTUDeque class

When deque is empty, method will be called as
        deque.iterator()
Size : 0 Element : []
Iterator will created successfully!

When deque has some elements, method will be called as
        deque.iterator()
Size : 5 Element : [A, B, C, D, E]
Iterator will created successfully!

D03 - Iterator<E> descendingIterator() method of GTUDeque class

When deque is empty, method will be called as
        deque.descendingIterator()
Size : 0 Element : []
Descending Iterator will created successfully!

When deque has some elements, method will be called as
        deque.descendingIterator()
Size : 5 Element : [A, B, C, D, E]
Descending Iterator will created successfully!

D04 - boolean add(E e) method of GTUDeque class

When deque is empty, method will be called respectively
         deque.add('X'), deque.add('Y'), deque.add('Z')
Before add method
Size : 0 Element : []
After add method
Size : 3 Element : [X, Y, Z]

When deque has some elements, method will be called respectively
         deque.add('K'), deque.add('L'), deque.add('M')
Before add method
Size : 3 Element : [X, Y, Z]
After add method
Size : 6 Element : [X, Y, Z, K, L, M]
```

```
D05 - void addFirst(E e) method of GTUDeque class

When deque is empty method will be called respectively as
        deque.addFirst('X'), deque.addFirst('Y'), deque.addFirst('Z')
Before addFirst method
Size : 0 Element : []
After addFirst method
Size : 3 Element : [Z, Y, X]

When deque has some elements, method will be called respectively as
        deque.addFirst('K'), deque.addFirst('L'), deque.addFirst('M')
Before addFirst method
Size : 3 Element : [Z, Y, X]
After addFirst method
Size : 6 Element : [M, L, K, Z, Y, X]

D06 - void addLast(E e) method of GTUDeque class

When deque is empty, method will be called respectively as
        deque.addLast('X'), deque.addLast('Y'), deque.addLast('Z')
Before addLast method
Size : 0 Element : []
After addLast method
Size : 3 Element : [X, Y, Z]

When deque has some elements, method will be called respectively as
        deque.addLast('K'), deque.addLast('L'), deque.addLast('M')
Before addLast method
Size : 3 Element : [X, Y, Z]
After addLast method
Size : 6 Element : [X, Y, Z, K, L, M]
```



```
D07 - boolean offer(E e) method of GTUDeque class

When deque is empty, method will be called respectively as
        deque.offer('X'), deque.offer('Y'), deque.offer('Z')
Before offer method
Size : 0 Element : []
offer 'X' result : true
offer 'Y' result : true
offer 'Z' result : true
After offer method
Size : 3 Element : [X, Y, Z]

When deque has some elements, method will be called respectively as
        deque.offer('K'), deque.offer('L'), deque.offer('M')
Before offer method
Size : 3 Element : [X, Y, Z]
offer 'K' result : true
offer 'L' result : true
offer 'M' result : true
After offer method
Size : 6 Element : [X, Y, Z, K, L, M]

D08 - boolean offerFirst(E e) method of GTUDeque class

When deque is empty, method will be called respectively as
        deque.offerFirst('X'), deque.offerFirst('Y'), deque.offerFirst('Z')
Before offerFirst method
Size : 0 Element : []
offerFirst 'X' result : true
offerFirst 'Y' result : true
offerFirst 'Z' result : true
After offerFirst method
Size : 3 Element : [Z, Y, X]

When deque has some elements, method will be called respectively as
        deque.offerFirst('K'), deque.offerFirst('L'), deque.offerFirst('M')
Before offerFirst method
Size : 3 Element : [Z, Y, X]
offerFirst 'K' result : true
offerFirst 'L' result : true
offerFirst 'M' result : true
After offerFirst method
Size : 6 Element : [M, L, K, Z, Y, X]
```

```
D09 - boolean offerLast(E e) method of GTUDeque class

When deque is empty, method will be called respectively as
        deque.offerLast('X'), deque.offerLast('Y'), deque.offerLast('Z')
Before offerLast method
Size : 0 Element : []
offerLast 'X' result : true
offerLast 'Y' result : true
offerLast 'Z' result : true
After offerLast method
Size : 3 Element : [X, Y, Z]

When deque has some elements, method will be called respectively as
        deque.offerLast('K'), deque.offerLast('L'), deque.offerLast('M')
Before offerLast method
Size : 3 Element : [X, Y, Z]
offerLast 'K' result : true
offerLast 'L' result : true
offerLast 'M' result : true
After offerLast method
Size : 6 Element : [X, Y, Z, K, L, M]

D10 - E element() method of GTUDeque class

When deque has some elements, method will be called respectively as
        deque.element()
Size : 5 Element : [A, B, C, D, E]
Element : A

D11 - E element() method of GTUDeque class

When deque is empty, method will be called respectively as
        deque.element(). Expected NoSuchElementException to be thrown
Size : 0 Element : []
Method will be called as deque.element() and
        NoSuchElementException was caught

D12 - E getFirst() method of GTUDeque class

When deque has some elements, method will be called respectively as
        deque.getFirst()
Size : 5 Element : [A, B, C, D, E]
Element : A

D13 - E getFirst() method of GTUDeque class

When deque is empty, method will be called respectively as
        deque.getFirst(). Expected NoSuchElementException to be thrown
Size : 0 Element : []
Method will be called as deque.getFirst() and
        NoSuchElementException was caught
```

```
D14 - E getLast() method of GTUDeque class

When deque has some elements, method will be called respectively as
        deque.getLast()
Size : 5 Element : [A, B, C, D, E]
Element : E

D15 - E getLast() method of GTUDeque class

When deque is empty, method will be called respectively as
        deque.getLast(). Expected NoSuchElementException to be thrown
Size : 0 Element : []
Method will be called as deque.getLast() and
        NoSuchElementException was caught

D16 - E peek() method of GTUDeque class

When deque is empty, method will be called respectively as
        deque.peek(). Expected return null
Size : 0 Element : []
Element : null

When deque has some elements, method will be called respectively as
        deque.peek()
Size : 5 Element : [A, B, C, D, E]
Element : A

D17 - E peekFirst() method of GTUDeque class

When deque is empty, method will be called respectively as
        deque.peekFirst(). Expected return null
Size : 0 Element : []
Element : null

When deque has some elements, method will be called respectively as
        deque.peekFirst()
Size : 5 Element : [A, B, C, D, E]
Element : A

D18 - E peekLast() method of GTUDeque class

When deque is empty, method will be called respectively as
        deque.peekLast(). Expected return null
Size : 0 Element : []
Element : null

When deque has some elements, method will be called respectively as
        deque.peekLast()
Size : 5 Element : [A, B, C, D, E]
Element : E
```

```
D19 - E remove() method of GTUDeque class

When deque has some elements, method will be called three times as
        deque.remove()
Size : 5 Element : [A, B, C, D, E]
Removed item : A
Removed item : B
Removed item : C
Size : 2 Element : [D, E]

D20 - E remove() method of GTUDeque class

When deque is empty, method will be called as
        deque.remove(). Expected NoSuchElementException to be thrown
Size : 0 Element : []
Method will be called as deque.remove() and
        NoSuchElementException was caught

D21 - E removeFirst() method of GTUDeque class

When deque has some elements, method will be called three times as
        deque.removeFirst()
Size : 5 Element : [A, B, C, D, E]
Removed item : A
Removed item : B
Removed item : C
Size : 2 Element : [D, E]

D22 - E removeFirst() method of GTUDeque class

When deque is empty, method will be called as
        deque.removeFirst(). Expected NoSuchElementException to be thrown
Size : 0 Element : []
Method will be called as deque.removeFirst() and
        NoSuchElementException was caught

D23 - E removeLast() method of GTUDeque class

When deque has some elements, method will be called three times as
        deque.removeLast()
Size : 5 Element : [A, B, C, D, E]
Removed item : E
Removed item : D
Removed item : C
Size : 2 Element : [A, B]
```

```
D24 - E removeLast() method of GTUDeque class

When deque is empty, method will be called as
        deque.removeLast(). Expected NoSuchElementException to be thrown
Size : 0 Element : []
Method will be called as deque.removeLast() and
        NoSuchElementException was caught

D25 - E poll() method of GTUDeque class

When deque is empty, method will be called as
        deque.poll(). Expected return null
Size : 0 Element : []
Element : null

When deque has some elements, method will be called three times as
        deque.poll()
Size : 5 Element : [A, B, C, D, E]
Element : A
Element : B
Element : C
Size : 2 Element : [D, E]

D26 - E pollFirst() method of GTUDeque class

When deque is empty, method will be called as
        deque.pollFirst(). Expected return null
Size : 0 Element : []
Element : null

When deque has some elements, method will be called three times as
        deque.pollFirst()
Size : 5 Element : [A, B, C, D, E]
Element : A
Element : B
Element : C
Size : 2 Element : [D, E]
```

D27 - E pollLast() method of GTUDeque class

```
When deque is empty, method will be called as
        deque.pollLast(). Expected return null
Size : 0 Element : []
Element : null

When deque has some elements, method will be called three times as
        deque.pollLast()
Size : 5 Element : [A, B, C, D, E]
Element : E
Element : D
Element : C
Size : 2 Element : [A, B]
```

D28 - boolean remove(Object o) method of GTUDeque class

```
When deque is empty, method will be called as
        deque.remove('X')
Size : 0 Element : []
remove 'X' result : false

When deque has some elements, method will be called respectively
        deque.remove('B'), deque.remove('X')
        deque.remove('D'), deque.remove('B')
Size : 5 Element : [A, B, C, D, E]
remove 'B' result : true
remove 'X' result : false
remove 'D' result : true
remove 'B' result : false
Size : 3 Element : [A, C, E]
```

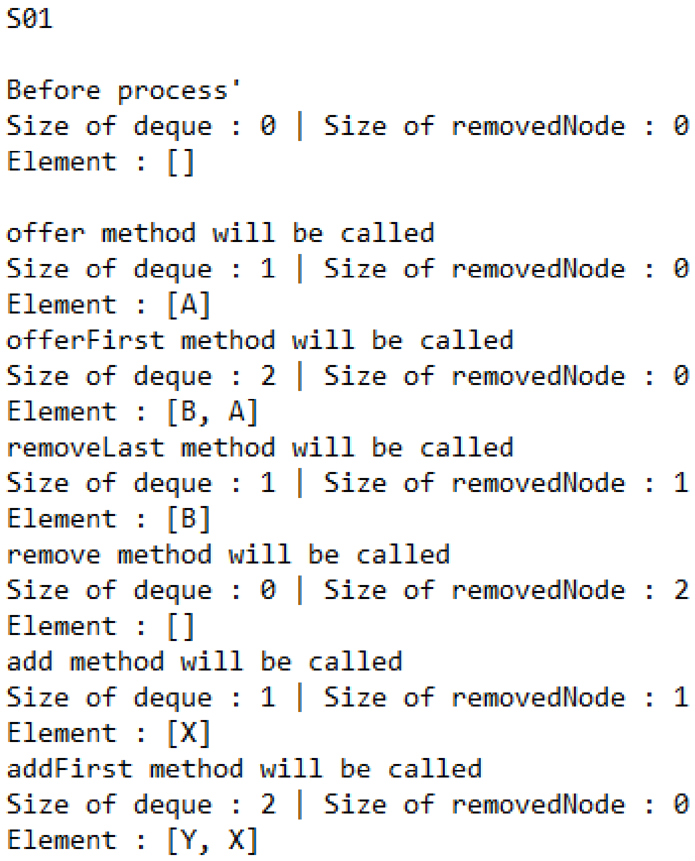
```
D29 - boolean removeFirstOccurrence(Object o) method of GTUDeque class

When deque is empty, method will be called as
        deque.removeFirstOccurrence('X')
Size : 0 Element : []
Remove 'X' result : false

When deque has some elements, method will be called
Size : 5 Element : [A, B, A, D, B]
removeFirstOccurrence 'D' result : true
Size : 4 Element : [A, B, A, B]
removeFirstOccurrence 'X' result : false
Size : 4 Element : [A, B, A, B]
removeFirstOccurrence 'A' result : true
Size : 3 Element : [B, A, B]
removeFirstOccurrence 'B' result : true
Size : 2 Element : [A, B]
removeFirstOccurrence 'A' result : true
Size : 1 Element : [B]
removeFirstOccurrence 'B' result : true
Size : 0 Element : []
removeFirstOccurrence 'B' result : false
Size : 0 Element : []

D30 - boolean removeLastOccurrence(Object o) method of GTUDeque class

When deque is empty, method will be called as
        deque.removeLastOccurrence('X')
Size : 0 Element : []
Remove 'X' result : false

When deque has some elements, method will be called
Size : 5 Element : [A, B, A, D, B]
removeLastOccurrence 'D' result : true
Size : 4 Element : [A, B, A, B]
removeLastOccurrence 'X' result : false
Size : 4 Element : [A, B, A, B]
removeLastOccurrence 'A' result : true
Size : 3 Element : [A, B, B]
removeLastOccurrence 'B' result : true
Size : 2 Element : [A, B]
removeLastOccurrence 'A' result : true
Size : 1 Element : [B]
removeLastOccurrence 'B' result : true
Size : 0 Element : []
removeLastOccurrence 'B' result : false
Size : 0 Element : []
```

```
D31 - boolean contains(Object o) method of GTUDeque class

When deque is empty, method will be called as
        deque.contains('X')
Size : 0 Element : []
Is it contain ? false

When deque has some elements, method will be called as
        deque.contains('C')
Size : 5 Element : [A, B, C, D, E]
Is it contain ? true

D32 - E pop() method of GTUDeque class

When deque has some elements, method will be called three times as
        deque.pop()
Size : 5 Element : [A, B, C, D, E]
Pop method return : A
Pop method return : B
Pop method return : C
Size : 2 Element : [D, E]

D33 - E pop() method of GTUDeque class

When deque is empty, method will be called as
        deque.pop()
Size : 0 Element : []
Method will be called as deque.pop() and
        NoSuchElementException was caught

D34 - void push(E e) method of GTUDeque class

When deque is empty, method will be called three times as
        deque.push('X'), deque.push('Y'), deque.push('Z')
Before push method
Size : 0 Element : []
After push method
Size : 3 Element : [Z, Y, X]

I01 - boolean hasNext() method of Iterator class

When deque is empty,
        iterator method will be called as iter.hasNext()
Size : 0 Element : []
hasNext method result : false

When deque has some elements,
        iterator method will be called as iter.hasNext()
Size : 5 Element : [A, B, C, D, E]
hasNext method result : true
```

```
I02 - E next() method of Iterator class

When deque has some elements,
        iterator method will be called three times as iter.next()
Size : 5 Element : [A, B, C, D, E]
next method result : A
next method result : B
next method result : C

I03 - E next() method of Iterator class

When deque is empty,
        iterator method will be called as iter.next()
Expected NoSuchElementException to be thrown
Size : 0 Element : []
Method will be called as iter.next() and
        NoSuchElementException was caught

I04 - void remove() method of Iterator class

When deque has some elements
        iter.next() method will be called
        then void remove() method will be called
Size : 5 Element : [A, B, C, D, E]
Next item : A
Next item : B
Next item : C
Size : 2 Element : [D, E]

I05 - void remove() method of Iterator class

When deque has some elements
        iter.remove() method will be called before iter.add() method
Expected IllegalStateException to be thrown
Size : 5 Element : [A, B, C, D, E]
Method will be called as iter.remove() and
        IllegalStateException was caught

DI01 - boolean hasNext() method of Iterator class

When deque is empty,
        iterator method will be called as iter.hasNext()
Size : 0 Element : []
hasNext method result : false

When deque has some elements,
        iterator method will be called as iter.hasNext()
Size : 5 Element : [A, B, C, D, E]
hasNext method result : true
```

```
DI02 - E next() method of Iterator class

When deque has some elements,
        iterator method will be called three times as iter.next()
Size : 5 Element : [A, B, C, D, E]
next method result : E
next method result : D
next method result : C

DI03 - E next() method of Iterator class

When deque is empty,
        iterator method will be called as iter.next()
Expected NoSuchElementException to be thrown
Size : 0 Element : []
Method will be called as iter.next() and
        NoSuchElementException was caught

DI04 - void remove() method of Iterator class

When deque has some elements
        iter.next() method will be called
        then void remove() method will be called
Size : 5 Element : [A, B, C, D, E]
Next item : E
Next item : D
Next item : C
Size : 2 Element : [A, B]

DI05 - void remove() method of Iterator class

When deque has some elements
        iter.remove() method will be called before iter.add() method
Expected IllegalStateException to be thrown
Size : 5 Element : [A, B, C, D, E]
Method will be called as iter.remove() and
        IllegalStateException was caught

M01

When deque has some elements,
        firstly iterator() method will be called
        then while iterator has next element, next method will be called
A B C D E F G H I J

M02

When deque has some elements,
        firstly descendingIterator() method will be called
        then while iterator has next element, next method will be called
J I H G F E D C B A
```

```
M03

When deque has some elements,
        firstly iterator() method will be called
        then while iterator has next element
        next and remove method will be called
Size : 5 Element : [A, B, C, D, E]
Next Item : A
Next Item : B
Next Item : C
Next Item : D
Next Item : E
Size : 0 Element : []

M04

When deque has some elements,
        firstly descendingIterator() method will be called
        then while iterator has next element
        next and remove method will be called
Size : 5 Element : [A, B, C, D, E]
Next Item : E
Next Item : D
Next Item : C
Next Item : B
Next Item : A
Size : 0 Element : []
Testing removed node size using adding or removing methods
```



```
S01

Before process'
Size of deque : 0 | Size of removedNode : 0
Element : []

offer method will be called
Size of deque : 1 | Size of removedNode : 0
Element : [A]
offerFirst method will be called
Size of deque : 2 | Size of removedNode : 0
Element : [B, A]
removeLast method will be called
Size of deque : 1 | Size of removedNode : 1
Element : [B]
remove method will be called
Size of deque : 0 | Size of removedNode : 2
Element : []
add method will be called
Size of deque : 1 | Size of removedNode : 1
Element : [X]
addFirst method will be called
Size of deque : 2 | Size of removedNode : 0
Element : [Y, X]
```