

Iterator worksheet

Iterator is an interface that has some method to retrieve elements from a collection object one by one. Let's use the Iterator interface to create a list of friends such that it iterates over the friends that begin with 'a'.

```
1. //worksheet for linkedlist and iterator
2. import java.io.*;
3. import java.util.*;
4. interface CSE12List<E>{
5.     public void insert(int index, E element);
6.     public void print();
7.     //other functions such as remove, find, etc
8. }
9. class FriendList<E> implements CSE12List<E>{
10.     //Inner class that is a node
11.     class Node{ //inner class
12.         E data;
13.         Node next;
14.         public Node(){
15.             data = null;
16.             next = null;
17.         }
18.         public Node(E data, Node before){
19.             if(before != null){
20.                 this.data = data; //assign data
21.                 this.next = before.next; //link to the element behind
22.                 before.next = this; //link from before
23.             }
24.         }
25.         public Node Next(){
26.             return next;
27.         }
28.     }
29.
30.
31.
32.
```

```

33.  class FriendListIterator implements Iterator { //inner class
34.      private int index;
35.      private Node left;
36.      private Node right;
37.      private boolean canRemove;
38.      //constructor for the iterator
39.      public FriendListIterator() {
40.          _____//initialize left
41.          _____//initialize right
42.          _____//iniitalize index
43.          _____//initialize canRemove
44.      }
45.      //override next method
46.      public boolean hasNext() {
47.          return _____//decision based on size
48.      }
49.      //next method
50.      public E next() {
51.          E result = null;
52.          if (size == 0){
53.              return null;
54.          }
55.          if (((String)right.data).startsWith("a")){//for exercise only
56.              result = right.data;
57.          }
58.          if (right.next != null){ //move to the next element
59.              _____
60.              _____
61.              _____
62.              _____
63.          }
64.          return result;
65.      }
66.  }

```

left	<input type="text"/>	right	<input type="text"/>
idx	<input type="text" value="0"/>		
canRemove	<input type="text" value="false"/>		

MyListlterator

67.

```
68.    //instance variables for FriendList
69.    private Node head;
70.    private Node tail;
71.    private int size;
72.
73.    //iterator method to get an iterator
74.    public Iterator<E> iterator() {
75.        return _____
76.    }
77.    //other methods for FriendList
78. }
```