Given the following Java code fragment:

```java
public class SomeClass {
    // Node type for linked list
    private class Node {
        private String item;
        private Node next;
        private Node (String value) { item = value; }
    }
    // first must always refer to first Node in the linked list
    private Node first = null;
}
```

For each operation:

1. **Insert a new Node with item = "B" into the linked list**

   **Before**
   
   ![Diagram](image1)

   **Java code**
   
   ```java
   Node temp = new Node("B");
   first = temp;
   ```

   **After**
   
   ![Diagram](image2)

2. **Insert a new Node with item = "A" into beginning of the linked list from (1)**

   **Before**
   
   ![Diagram](image3)

   **Java code**
   
   ```java
   Node temp = new Node("A");
   temp.next = first;
   first = temp;
   ```

   **After**
   
   ![Diagram](image4)
3. Insert a new `Node` with `item = "D"` at the end of the linked list from (2)

<table>
<thead>
<tr>
<th>Before</th>
<th>Java code</th>
<th>After</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

4. Insert a new `Node` with `item = "C"` after the `Node` with `item = "B"` in the list from (3)

<table>
<thead>
<tr>
<th>Before</th>
<th>Java code</th>
<th>After</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

5. Using a do-while loop, print all the items in the linked list from (4).

```java
if (first == null) 
    StdOut.println("EMPTY LIST");
else 
    Node current = first; 
    do { 
        
    while ( 
    ) ;
```

6. Suppose you have a linked list. The start of the list is stored in `Node first`. Write a loop to add a new `Node` with `item = "E"` to the end of linked list. The list may contain 0 or more items.