1 PandaList

Reference the class definitions below:

```java
public class Panda {
    static String food;
    int age; // any int
    boolean happy;

    public Panda(int age) {
        this.age = age;
    }
}

public class PandaList {
    public Panda item;
    public PandaList rest;

    public PandaList(Panda p) {
        this.item = p;
        this.rest = null;
    }
}
```

What output will result from executing the following main method? Note, we strongly suggest you draw a box and pointer diagram.

```java
public static void main(String[] args) {
    Panda p = new Panda(1);
    PandaList lst = new PandaList(p);
    lst.rest = new PandaList(p);
    lst.rest.rest = lst;
    p = new Panda(-10);
    PandaList temp = lst;
    lst = new PandaList(p);
    lst.rest = temp;
    lst.rest.rest.item.food = "Bamboo";
    temp.rest.rest.item.happy = true;
    temp.item.food = "shoots";
    lst.rest.rest = lst;
    lst.rest.rest.rest.item.happy = false;
    lst.rest.rest.item.happy = true;
    System.out.println(temp.rest.item.age); // Print statement 1
```
System.out.println(temp.rest.rest.rest.item.food); // Print statement 2
System.out.println(lst.rest.item.happy); // Print statement 3
System.out.println(temp.rest.rest.item.happy); // Print statement 4
System.out.println(lst.rest.rest.item.age); // Print statement 5
}

For each of the following sub-questions, select what is printed out by the corresponding print statements.

Print Statement One: ( ) -10 ( ) 1 ( ) false

Print Statement Two: ( ) Bamboo ( ) shoots ( ) null

Print Statement Three: ( ) true ( ) false

Print Statement Four: ( ) true ( ) false

Print Statement Five: ( ) -10 ( ) 1 ( ) false
2 Linked List Potpourri

Reminder: constant-time here means that regardless of the length of the list, this operation will take the same amount of time.

**Part 1**  An encapsulated **singly** linked list with a pointer to **only** the first element allows for which of the following methods to be constant-time operations (assuming a reasonable implementation)?

- [ ] addFirst(int x)
- [ ] addLast(int x)
- [ ] removeFirst()
- [ ] removeLast()
- [ ] remove(Node n) (destructively removes Node n from the list)

**Part 2**  An encapsulated **singly** linked list with a pointer to **both** the first element and the last element allows for which of the following methods to be constant-time operations (assuming a reasonable implementation)?

- [ ] addFirst(int x)
- [ ] addLast(int x)
- [ ] removeFirst()
- [ ] removeLast()
- [ ] remove(Node n) (destructively removes Node n from the list)

**Part 3**  An encapsulated **doubly** linked list with a pointer to **both** the first element and the last element allows for which of the following methods to be constant-time operations (assuming a reasonable implementation)?

- [ ] addFirst(int x)
- [ ] addLast(int x)
- [ ] removeFirst()
- [ ] removeLast()
- [ ] remove(Node n) (destructively removes Node n from the list)