Sejal Jain

sj2735

COMSW3134 HW 2

Problem 1

```
public void printLots(List<AnyType> L, List<AnyType> P) {
        Iterator<Integer> iter = P.iterator();
        while (iter.hasNext()) {
                System.out.println(L[iter.next()]);
        }
}
```

Problem 2

```
public List<AnyType> printLots(List<AnyType> L1, List<AnyType> L2) {
        List<AnyType> result;
        Iterator<Integer> iter = L1.iterator();
        while (iter.hasNext()) {
                if L2.contains(iter.next()) {
                        result.add(iter.next());
                }
        }
}
```

Problem 3

```
int arr_size = 500;
int[] arr = new int[arr_size];
stack1_top = 0;
stack2_top = arr_size-1;
public void stack1Push(int x) {
        if (!overflow()) {
                arr[stack1_top] = x;
                Stack1_top++;
        }
        else {
                System.out.println("Overflow");
        }
}
public void stack2Push(int x) {
        if (!overflow()) {
                arr[stack2_top] = x;
                stack2_top--;
        }
```

```
        else {
                System.out.println("Overflow");
        }
}
public int stack1Pop() {
        stack1_top--;
        return arr[stack1_top];
}

public int stack2Pop() {
        stack2_top++;
        return arr[stack1_top];
}
public int stack1Top() {
        return arr[stack1_top - 1];
}
public int stack2Top() {
        return arr[stack1_top + 1];
}
public boolean overflow() {
        if(stack1_top > stack2_top) {
                return true;
        }
        else {
                return false;
        }
}
```

Problem 4

Part 1

1. Move 4 to S1
2. Move 3 to S2
3. Move 1 to output
4. Move 8 to S3
5. Move 2 to output
6. Move 3 to output
7. Move 4 to output
8. Move 7 to S1
9. Move 6 to S1
10. Move 9 to S2
11. Move 5 to output
12. Move 6 to output
13. Move 7 to output

14. Move 8 to output
15. Move 9 to output

Part 2

1 9 8 7 6 5 4 3 2