

1.

```

public static void printLots(List<Integer> L, List<Integer> P){
    Iterator<Integer> li = L.iterator();
    Iterator<Integer> pi = P.iterator();
    int index = 0;
    int target;

    while(pi.hasNext()){
        target = pi.next();
        while(index!=target&&li.hasNext()){
            index++;
            if (li.hasNext()) {
                li.next();
            }
            else{
                System.out.println("Index out of range.");
            }
        }
        System.out.println(""+li.next());
        index++;
    }
}

```

3.4 intersection problem

```

List solution = new List();
Iterator one = L1.iterator();
Iterator two = L2.iterator();
L1Item = one.next();
L2Item = two.next();
while(L1 and L2 both hasNext()){
    if(L1Item>L2Item){
        L2Item = two.next();
    }
    else if(L2Item>L1Item){
        L1Item = one.next();
    }
    else{
        solution.add(L1Item)
        one.next();
        two.next();
    }
}
print(solution)
print(solution)

```

```

3.24 public void push(item){
    if (stack1.size + stack2.size > array.length-1) {
        declareOverflow();
    }
    else if(pushing on stack1){
        array[stack1.size] = item;
    }
}

```

```
        else if(pushing on stack2){
            array[array.length-stack2.size-1] = item;
        }
    }
    public AnyType pop(){
        if(popping from stack1){
            stack1.size -=1;
            return array[stack1.size];
        }
        if(popping from stack2){
            stack2.size-=1;
            return array[array.length-stack2.size-1]
        }
    }
}
```

#### 4. Train problem

a.)

```
train 4 from input to S1
train 3 from input to S2
train 1 from input to output
train 8 from input to S3
train 2 from input to output
train 3 from S2 to output
train 4 from S1 to output
train 7 from input to S1
train 6 from input to S3 (on top of 8)
train 9 from input to S2
train 5 from input to output
train 6 from S3 to output
train 7 from S1 to output
train 8 from S3 to output
train 9 from S2 to output
```

b.)

[4, 5, 9, 8, 7, 6, 3, 2, 1]

this cannot be solved because of the 6-7-8-9 portion, you need to put 6, 7, and 8 on their own holding tracks because you can't stack a larger number train on top of a smaller number train. Once 6 7 and 8 are all on their own tracks, there is nowhere for train 9 to go because it is larger than all three of these trains.