Zoe Gordin
zeg2103
10/10/16

1. You are given a list, L , and a another list, P, containing integers sorted in ascending order. The operation printLots(L,P) will print the elements in L that are in positions specified by P. For instance, if P=1,3,4,6, the elements in positions 1,3,4, and 6 in L are printed. Write the procedure printLots(L,P). The code you provide should be the java method itself (not pseudocode), the containing class is not necessary. You may use only the public Collection methods that are inherited by lists L and P. You may not use methods that are only in List. Anything that the Collection methods return is fair game, so you might think about how to use iterators.

```
Int variables pPrior, lNext, and pNext
P and L iterators
pPrior = 0, lNext = 0
while(pIterator.hasNext())
{
        pNext = pIterator.next();
        for(int i = 0; i<pNext - pPrior; i++)
        {
                lNext = lIterator.next();
        }
        pPrior = pNext;
        System.out.println(lNext);
}
```

1. 3.4 Given two sorted lists, L1 and L2, write a procedure to compute L1 ∩ L2 using only the basic list operations.

```
List<AnyType> L3 = new List<AnyType>();
For each object in L1
        If the object in L1 also exists in L2:
                Add the object to L3
```

2. 3.24 Write routines to implement two stacks using only one array. Your stack routines should not declare an overflow unless every slot in the array is used

   Stack class with 2 of each of the typical stack methods:
   Push1 pushes an object to the first stack
   Push2 pushes and object to the second stack
   Pop1 pops from the first stack, pop2 pops from the second stack
   Peek1 shows the top item of the first stack, peek2 shows the top item of the second stack

   The array holds the stacks by one stack filling the array from the first element of the array, and the second stack fills the array from the last element of the array. To check for overflow, examine the amount of elements in between the two top elements of the stacks within the array.

4a.

1. S14 S2_ S3_ output:
2. S14 S23 S3_ output:
3. S14 S23 S3_ output: 1
4. S14 S23 S38 output: 1
5. S14 S23 S38 output: 2 1
6. S14 S2_ S38 output: 3 2 1
7. S1_ S2_ S38 output: 4 3 2 1
8. S17 S2_ S38 output: 4 3 2 1
9. S16,7 S2_ S38 output: 4 3 2 1
10. S16,7 S29 S38 output: 4 3 2 1
11. S16,7 S29 S38 output: 5 4 3 2 1
12. S17 S29 S38 output: 6 5 4 3 2 1
13. S1_ S29 S38 output: 7 6 5 4 3 2 1
14. S1_ S29 S3_ output: 8 7 6 5 4 3 2 1
15. S1_ S2_ S3_ output: 9 8 7 6 5 4 3 2 1

4b. Example: [1,2,3,7,8,9,4,5,6]