HW-3

1.

Write each of the following *pure* functions (a function is pure if it does not update any of the arguments or global variables). No functions in the `Collections` class can be used.

1.1 The function `take(n, lst)` returns a list that takes `n` elements from `lst`.

```
public static <E> LinkedList<E> take(int n, LinkedList<E> lst)
```

1.2 The function `drop(n, lst)` returns a list that keeps all the elements of `lst` except for the first `n` elements. For example, `drop(5,[1,2,3,4,5,6,7,8,9,10])` returns `[6,7,8,9,10]`.

```
public static <E> LinkedList<E> drop(int n, LinkedList<E> lst)
```

1.3 The function `reverse(lst)` returns a copy of the list `lst` with the elements reversed.

```
public static <E> LinkedList<E> reverse(LinkedList<E> lst)
```

1.4 The function `sortedDown(lst)` checks if `lst` is sorted in *non-increasing* order. For example, for `lst = [3,3,2,1]`, it returns true.

```
public static <E extends Comparable<E>> boolean sortedDown(LinkedList<E> lst)
```

2.

Write a program that reads words from a text file and displays all the words (duplicates allowed) in ascending alphabetical order. The text file is passed as a command-line argument.

3.

A Java program contains various pairs of grouping symbols, such as:

- Parentheses: (and)
- Braces: and
- Brackets: [and]

Note that the grouping symbols cannot overlap. For example, `(a{b)}` is illegal. Write a program to check whether a Java source-code file has correct pairs of grouping symbols. Pass the source-code file name as a command-line argument.

4.

The *heap-sort* algorithm sorts a collection using the heap data structure. Since the heap data structure is used in the implementation of `PriorityQueue`, you can implement the heap-sort algorithm using `PriorityQueue`. Do the implementation:

```
public static <E extends Comparable<E>> LinkedList<E> heapSort(LinkedList<E> lst)
```

Notice that this is a pure function.

5. (project)

Write a method that converts an infix expression into a postfix expression using the following header:

```
public static String infixToPostfix(String expression)
```

For example, the method should convert the infix expression `(1 + 2) * 3` to `1 2 + 3 *` and `2 * (1 + 3)` to`2 1 3 + *`.

6. (project)

The program shown below counts the occurrences of words in a text, and displays the words and their occurrences in alphabetical order. Suppose you have a dictionary that maps words of different forms to their root forms. For example, the dictionary maps verbs *see*, *sees*, *seeing*, *saw*, and *seen* to their root form *see*. Write a program that counts the occurrences of words in a text, treating all forms of a root word as the root word.

```
import java.util.*;

public class CountOccurrenceOfWords {
  public static void main(String[] args) {
    // Set text in a string
    String text = "Good morning. Have a good class. " +
      "Have a good visit. Have fun!";

    // Create a TreeMap to hold words as key and count as value
    Map<String, Integer> map = new TreeMap<>();

    String[] words = text.split("[\\s+\\p{P}]");
    for (int i = 0; i < words.length; i++) {
      String key = words[i].toLowerCase();

      if (key.length() > 0) {
        if (!map.containsKey(key)) {
          map.put(key, 1);
        }
        else {
          int value = map.get(key);
          value++;
          map.put(key, value);
        }
      }
    }

    // Display key and value for each entry
    map.forEach((k, v) -> System.out.println(k + "\t" + v));
  }
}
```

7. (optional)

- Write a function that returns the number of zeros in a given simple list of numbers.
- Write a function that takes a simple list of numbers as a parameter and returns a list with the largest and smallest numbers in the given list.
- Write a function that takes an integer n as a parameter and prints the first n rows of the Pascal's triangle.
- Arrays: Write a program that tests if a 2D square array is symmetric about the diagonal from (0,0) to (n-1,n-1). (from EPI)
- Stacks and Queues: Write a programt to evaluate arithmetical expressions that use + and * applied to nonnegative integer arguments. Expressions are in reverse-Polish notation, e.g., 3 4 + 5 *, 1 3 + 5 7 + *. (from EPI)
- Hash tables: Write a program that finds the most common object in an array of objects. Each object is a pair of strings. Treat strings as being the same if they are equal when converted to lower case. (from EPI)
- Write a program to find the unique positive integer whose square has the form `1_2_3_4_5_6_7_8_9_0`, where each `"_"` is a single digit. (from Euler)