Exercise of Programming Language, Quiz E5

Write 4 Python programs to solve the following questions. Please name your program files as *Q1.py*, *Q2.py*, and so on, *i.e.*, according to the serial number of questions.

1. Compute how many times each “word” occurs in the following sentence.

   ```
   "If two witches would watch two watches, which witch would watch which watch?"
   ```

   Please sort your output according to the alphabet sequence of the words (separated with TAB), which should look like this:

   ```
   Word    Occurrence
   If      1
   two     2
   watch   3
   watches 1
   which   2
   witch   1
   witches 1
   would   2
   ```

2. Write a program that **asks the user to input an integer** and prints to your screen all the prime numbers (separated with TAB) and the number of prime numbers smaller or equal to the given integer. Please use the **while** loop; for loop is forbidden.

   Output example:

   ```
   Please input a number: 50
   Prime number <= 50 are listed here:
   2   3   5   7   11   13   17   19   23   29   31   37   41   43   47

   Number of primes: 15
   ```

3. Please use a **recursion function** to compose a program that finds the factorial number (*n*!) of *n*.

   ```
   assert factorial(5) == 120
   assert factorial(10) == 3628800
   ```

4. The selection sort improves on the bubble sort by making only one exchange for every pass through the list. In order to do this, a selection sort looks for the largest value as it makes a pass and, after completing the pass, places it in the proper location.

   As with a bubble sort, after the first pass, the largest item is in the correct place. After the second pass, the next largest is in place. This process continues and requires *n–1* passes to sort *n* items, since the final item must be in place after the $(n-1)^{th}$ pass.

   Write a program to present the step of iteration of sorting process.

   Sample Input:

   ```
   Numbers = [54, 26, 93, 17, 77, 31, 44, 55, 20]
   ```

   Sample Output:

   ```
   [54, 26, 20, 17, 77, 31, 44, 55, 93]
   [54, 26, 20, 17, 55, 31, 44, 77, 93]
   [54, 26, 20, 17, 44, 31, 55, 77, 93]
   [31, 26, 20, 17, 44, 54, 55, 77, 93]
   [31, 26, 20, 17, 44, 54, 55, 77, 93]
   [17, 26, 20, 31, 44, 54, 55, 77, 93]
   [17, 20, 26, 31, 44, 54, 55, 77, 93]
   [17, 20, 26, 31, 44, 54, 55, 77, 93]
   [17, 20, 26, 31, 44, 54, 55, 77, 93]
   ```

Hint: this figure shows the entire sorting process