This exam consists of 6 questions worth a total of 100 points. Make sure your copy has all 6 questions. The points available for each question are indicated to the left of the question.

1. (10) Which of the following is NOT a generally a feature of object-oriented design?
   (a) Organizing data and methods in classes.
   (b) Identifying abstract data types that will be represented as classes.
   (c) Start with a high-level description of what the program will do and continue to break the problem into smaller pieces with each piece assigned a function.
   (d) Use of inheritance
   (e) Design test code for each class.

2. (20) A palindrome is a string that reads the same both forward and backwards. An example palindrome is the string "able was i ere i saw elba"
   Define a recursive algorithm to determine if a string is a palindrome.

Questions 3-6 are all to be considered parts of a single question about a polynomial class.
Consider the “SortedList” class whose data and methods are listed below. A SortedList is one that inserts new nodes into the list in its proper (sorted) location. All operations on a sorted list ensure that the resulting list stays sorted.

```
Class SortedList {
    protected link head;
    private link tail;

    protected void insert(int newNode)
    {
    }

    protected int removeSmallest()
    {
    }

    protected int removeLargest()
    {
    }
}
```
A polynomial is a summation of terms, each of which takes the form of:

\[ a_k x^k \]

Thus, an example polynomial is:

\[ 47x^{19} + 6x^{14} - 3x^9 + 7 \]

By convention, the terms of polynomials are listed in decreasing order of the exponent value, as shown in the example above.

3. (20) We first need to “design” the polynomial class. Using a format similar to that shown above for SortedList, include the basic arithmetic functions necessary for a polynomial (+, −, ∗, /). Also, show or describe how you would make use of the SortedList class in your polynomial class.

4. (20) Implement the polynomial class. Show the java code “here.”

5. (10) Write a driver to test the polynomial class.

6. (20) To make matters simpler, polynomials typically are shown with like terms (those with the same exponent) combined. The following shows an example. Assuming that during manipulation of polynomials you wind up with one that looks like:

\[ 47x^{19} - 5x^{19} + 6x^{14} + 2x^{14} - 3x^9 + 7 \]

We would generally combine the \( x^{19} \) and \( x^{14} \) terms leaving

\[ 42x^{19} + 8x^{14} - 3x^9 + 7 \]

Given your current design and implementation of the polynomial class, how would you change your design and implementation to support combining of like terms in a polynomial.