CS 200 Lab 8

Problem 1: (30)
Write **a function called mergeLists** that takes two call-by-reference arguments that are pointer variables that point to the *ascending sorted* arrays of values of type int. The function returns a pointer to a new *sorted* array that contains all of the nodes in the original two lists. Also, the TA's should be able to enter values into the array for testing purposes.

Example:

a = 1, 2, 8
b = 3, 7, 35, 100

mergeLists(a, b)
=
1, 2, 3, 7, 8, 35, 100

Problem 2: (70)

Using dynamic arrays, implement **a polynomial class** with polynomial addition, subtraction, and multiplication. A polynomial will be a string in this format : "8x4+3x2+2"
A variable in a polynomial does very little other than act as a placeholder for the coefficients. Hence, the only interesting thing about polynomials is the array of coefficients and the corresponding exponent. Think about the polynomial "x*x*x + x + 1". One simple way to implement the polynomial class is to use an **array of doubles** to store the coefficients. The index of the array is the exponent of the corresponding term. Where is the term in x*x in the previous example? If a term is missing, then it simply has a zero coefficient.

Provide a default constructor, a copy constructor, and a parameterized constructor that enables an arbitrary polynomial to be constructed. Also supply an overloaded operator = and a destructor.

Provide **these operations:**
■ polynomial + polynomial
■ polynomial - polynomial
■ polynomial * polynomial
■ constant * polynomial
■ polynomial * constant

You are supposed to overload operators for each of these operations.

- Supply a function to evaluate the polynomial at a value of type double .
- You should decide whether to implement these functions as members, friends, or standalone functions.