Question 6: The addDecreasingOrder method assumes that the list is ordered in decreasing integer order. The method enables adding elements to the list while maintaining this order. Example: if this list is (12 9 3) and we now add x=5 to the list, this list becomes (12 9 5 3). Another example: If this list is (12 9 3) and we now add x=17, this list becomes (17 12 9 3). Implement the method.

/\*\* Assumes that this list is sorted in decreasing order.

\* Adds the given element to this list, while maintaining the list's order.\*/

public void addDecreasingOrder(int x) {

Question 8: Given an int value, say 140703, we wish to construct from its digits the biggest possible integer. In this example the answer is 743100. Hint: this problem can be solved elegantly by using some of the services of the LinkedList class. Implement the following method, using this hinit. You are not allowed to use String conversions. Rather, you have to operate directly on the given integer.

/\*\* Returns the biggest integer that can be constructed from the digits of

\* the given integer. For example, if x = 231, returns 321. \*/

public static int biggestNumber(int x) {

// Write your code here:

Question 12: The program half reads a number from the input, let’s call this number x, and writes to the output the integer part of x/2. For example, if x is 8, the program writes 4. If x is 11, the program writes 5. If x is 0 or 1, the program writes 0. Implement the program below, using the Vic symbolic language. In other words, use commands like store x, goto LOOP, etc. Assume that the Vic assembler will translate your program into executable code.

// Reads a number, say x, and outputs the integer part of x/2.

// Write your symbolic Vic code below.

Question 10: The following function checks if one array is a sub-array of another array. Example: any one of the following arrays is a sub-array of the array [7 3 8 1 6 9 2 4]:  
[3 6 2], [8], [3 6 2 4], [7 8 4], [] (empty array), to give some examples. The array [3 2 6] is *not* a sub-array of this array, because it’s not in the same order.

Write a recursive solution. If you want, you can write another private method. A non-recursive solution is also acceptable, for 3 points less.

/\*\* Returns true if the first array is a sub-array of the second array. \*/

public static boolean subArr (int[] arr1, int[] arr2) {

// Write your code below: