Homework 1 (20 pts)

Problem 1: Sparse Vector Using Linked List (7 pts)

- Implement the linked list sparse vector class (LLSparseVec.java) so that LLMainClass can be executed.
- Nodes in the linked list are nonzero elements of the vector, sorted according to their index (refer to the slide in Lecture 2 – part 2).
- Implement the constructor, access methods, getElement, setElement, clearElement, getAllIndices, getAllValues. In otherwords, when LLMainClass is called using VEC argument and with a single input file, the program should be able to run correctly and give the same output as ArrayMainClass.

Problem 2: Sparse Vector Operation Linked List (7 pts)

- Implement the addition, substraction, and multiplication methods in LLSparseVec.
- The algorithm has to be O(m), in which m is the maximum number of nonzero elements in a vector. To achieve this, you cannot simply use get and set in Problem 1. Only algorithmes with O(m) complexity will get credits.
- All operations return a new sparse vector object, storing the result. The method substraction(otherV) means the current vector minus otherV.
- If the two vectors' length do not match, return a null object.
- When LLMainClass is called using VEC argument and with multiple input files, the program should be able to run correctly and give the same output as ArrayMainClass.

-