We have seen in class how selection sort algorithm works on arrays data structure. In this lab we will practice how selection sort can be performed on a linked list ADT.

1. Convert the following selection sort pseudo-code to perform the sort in ascending order.
   (selectionSort_asc function)
   a. Find the node with the minimum value in the linked list of length n
   b. Append the min node in a new result linked list
   c. Delete min from original linked list
   d. Repeat steps a-c until the original linked list is empty
   e. Return the result linked list

2. Convert the following selection-sort pseudo-code to perform the sort in descending order.
   (selectionSort_desc function)
   a. Find the node with the maximum value in the linked list of length n
   b. Append the max node in a new result linked list
   c. Delete max from original linked list
   d. Repeat steps a-c until the original linked list is empty
   e. Return the result linked list

Note: You might assume that all characters are lower-case.

For example:

Input: str = “ilovedata”

Output: -> a -> a -> d -> e -> i -> l -> o -> t -> v (using selectionSort_asc)
Output: -> v -> t -> o -> l -> i -> e -> d -> a -> a (using selectionSort_desc)
printList(head);
head = selectionSort_asc(head);
System.out.println("\n List After selectionSort_asc");
printList(head);

// Expected answer: -> a -> a -> d -> e -> i -> l -> o -> t -> v

head = initializeList("ilovedata");
System.out.println("\n List Before selectionSort_desc");
printList(head);
head = selectionSort_desc(head);
System.out.println("\n List After selectionSort_desc");
printList(head);

// Expected answer: -> v -> t -> o -> l -> i -> e -> d -> a -> a

public static Node selectionSort_asc(Node head){
    Node result = null;
    // INSERT CODE HERE

    return result;
}

public static Node selectionSort_desc(Node head){
    Node result = null;
    // INSERT CODE HERE

    return result;
}

// Method that takes a string and insert its characters into a linked list
public static Node initializeList(String str){
    Node head = new Node(str.charAt(0)), cur;
    int i;

    for(cur= head,i=1;i<str.length();i++,cur=cur.next){
        cur.next = new Node(str.charAt(i));
    } return head;
}

// Method for printing linked list
public static void printList(Node head){
    Node cur = head;
    if(head==null) System.out.print("<EMPTY>");
    for(;cur!=null;cur=cur.next){
        System.out.print("-> "+cur.item+" ");
    }
}
}