COMP5112 Lab 2

1. Background: Linked List

1) In a linked list, each Item *x* stores:

   x.key; *x*.prev; *x*.next:

   The constructor function of *Item* is:

```
Item(int k) {
        key=k;
        prev=null;
        next=null;
    }
```

   To create an item, the statement can be:

   Item *a* = new Item(3);

2) In a linked list, the first item in the link is represented as *head*. The constructor function of Linked List is:

```
LinkedList() {
        head = null;
    }
```

   To create a Linked List, the statement can be:

   LinkedList *L*=new LinkedList();

3) Linked List *L* contains some functions:

   1) *L*.Search(int *k*)

      To search an item with the key value *k* in Linked List *L*

   2) *L*.Insert(Item *x*)

      To insert an item *x* into Linked List *L*

   3) *L*.Delete(Item *x*)

      To delete an item *x* from Linked List *L*

   4) *L*.Print()

      To print all the keys in the Linked List *L*

   5) *L*.Reverse()

      To reverse the order of items in the Linked List *L*

2. Questions

Question 1. Linked List Print()

Compile the "Lab2.java" file, fill in the blank indicated by **//1: _____** and test it using TestPrint() function. The expected output:

Contents in the linked list: [1 8 5 2 3]

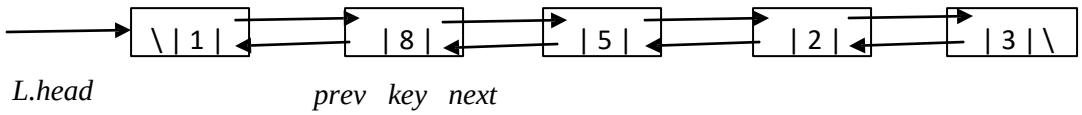
Question 2. Linked List Reverse()

Compile the "Lab2.java" file, fill in the blanks indicated by **//2: _____** and test it using TestReverse() function. The expected output:

Contents in the linked list: [3 2 5 8 1]

Question 3. Stack Simulation (Exercise)

Now, we have the implementation of Linked List (in Lab2.java) and we can get a Linked List *L* as follows:



A linked list *L* supports these operations:

Insert(Item *x*): To insert an item *x* into Linked List *L*

Delete(Item *x*): To delete an item *x* from Linked List *L*

Now, we want to implement a stack *S* by using a linked list *L*.

Recall that a stack *S* supports two operations:

Push(*S*, *x*): to insert *x* to the top of the stack

Pop(*S*): to extract the top item

How can we implement these two operations (for a stack *S*) by using a linked list *L*?

Hints: How can we implement the Push operation by using the Insert operation?

How can we implement the Pop operation by using the Delete operation?