

Objective(s) :

- To understand the basic implementation of a singly linked list
- Be able to manipulate list structure.

task 1: Implement MyLinkedList.Java, inside package solutions.code3 including all the method mentioned in lectures, i.e.

public void add(int d) -> add a node with value d at the head of the linked list

public void insert(int d) -> insert into **an ordered linked list** (ascendingly)

public int find(int d) -> return the index of the node valued d, or -1 if not found.

public void delete(int d) -> delete from a linked list

Note that It's common to have a clearly defined method append(int d) for keeping natural input order to the linked list.

```
static void demo1() {  
    MyLinkedList list = new MyLinkedList();  
    list.add(5);  
    list.add(1);  
    list.insert(4);  
    list.insert(3);  
    System.out.println(list);  
    list.delete(2);  
    System.out.println("5 is at " + list.find(5));  
    System.out.println(list);  
}
```

-demo1-----

head->(1)->(3)->(4)->(5)->null

5 is at 3

task 2: Implement the following methods (for task3)

public int size() -> number of elements in the list

public void add(int [] d) -> add to the list with values from d[length - 1] to d[0] i.e. reverse the order from array d because add(int d) inserts d[i] to the front which makes its content reversed order from the input

public void insert(int [] d) -> add to the list with values from d[i]. Since insert always result in ordered list, it should simply call insert(d[i])

```
static void demo2() {  
    int [] arr = {50,40,30,20,10};  
    MyLinkedList mList = new MyLinkedList();  
    mList.insert(arr);  
    System.out.println(mList);  
}
```

-demo2-----

head->(10)->(20)->(30)->(40)->(50)->null

submission: MyLinkedList_XXYYYY.java where XX is the first 2 digit and YYYY is the last 4 digit of your student id.

Due date: TBA