Shoubra Faculty of Engineering **Computer Engineering Department**
Benha University **3rd year**

Data Structure: Assignment #3

Programming problems:

1. Write and test this method:

   void put (LinkList list, int i, int x)

   // inserts x as element number i in the double-ended linked list L;

   For example, if list is {22, 33, 44, 55, 66, 77, 88, 99}, then put (list, 3, 50) will change list to {22, 33, 44, 50, 55, 66, 44, 88, 99}. Hint: if i = 0, replace the value of the first Linkwith x, and insert a new Linkimmediately after it that contains the previous fist value.

Soln:

```
void put (LinkList list, int i, int x) {
// inserts x as element number i;
if (list.first == null) {
System.out.println("list is empty"); }
else if (i == 0)
{
Link p = new Link(x);
p.next = list.first;
list.first= p;
}
 else
{
Link p = list.first;
int  j = 1;
while (j < i && p != null)
{
++j;
p = p.next;
```

```
}
if (p == null)
{
String error=String.format("the list has only %d elements",j-1);
throw new java.util.NoSuchElementException(error);
}
Link t=new Link(x);
t.next=p.next;
p.next =t;
if (p == list.last)
   list.last=t;
}
}
```

2. Write and test this method:

void swap (LinkList list, int i, int j)

// swaps the ith element with the jth element in the doubly linked list L;

For example, if list is {22, 33, 44, 55, 66, 77, 88, 99}, then swap (list, 2, 5) will change list to {22, 33, 77, 55, 66, 44, 88, 99}.

Soln:

```
void swap(LinkList list, int i, int j) {
if (i < 0 || j < 0) {
throw new IllegalArgumentException();
} else if (i == j) {
return;
}
Link p=list.first, q=list.first;
for (int x=0; x<i; x++) {
if (p == null) {
throw new IllegalStateException();
}
p = p.next;
}
for (int y=0; y<j; y++) {
if (q == null) {
throw new IllegalStateException();
}
q = q.next;}
```

```
int pdata = p.data, qdata = q.data;

p.data = qdata;

q.data = pdata;

return;

}
```

Assignment Problem:

- Write and test this method:

void rotateLeft (Linklist)

// moves the first element of the specified list to its end; For example, if list is {22, 33, 44, 55, 66, 77, 88, 99}, then rotateLeft (list) will change list to {33, 44, 55, 66, 77, 88, 99, 22}. Note that no new nodes are created by this method.