南京大學

1、(a)、交换 cur 和 cur.next

```
Node pnext = cur.next
cur.next = cur.next.next
pnext.next = cur
pre .next = pnext
```

(b)、交换 cur 和 cur.next

```
Node pnext = cur.next
cur.next = pnext.next
pnext.next.ahead = cur
bef.next = pnext
pnext.ahead = bef
pnext.next = cur
cur.ahead = pnext
```

2、

```
Iterator<int> i1 = List1.iterator();
Iterator<int> i2 = List2.iterator();
while (i1.hasNext())
  int tem = i1.next;
  while (i2.hasNext())
    if( i2.next() == tem ) ;
    System.out.println(i2.next());
      break;
```

] - ①

3.

```
for(int i : List())
  System.out.println(i);
```

① 代码改, 将 i2.next() == tem 改为 i2.next() != tem

4

```
LinkedList<int> L1 = new LinkedList();
Node p = L.head();
while(p != null){
Node pnow = p.next;
p.next = L1;
L1 = p;
p = next;
}
```

1. (1). B

(2). C

2. (1). Node tem = p

p = x

p.next = tem

(2). p = p.next

3. (1). (rear + 1) % length

(2). length == 0;

length == m;

```
Enqueue(x){
if( length <= m-1 ){
  for(int i = rear; i <= length; i++){
    q[i] = q[i-1]; }
  q[rear] = x;
  }
else throw new BoundaryHitOfException();

dlqueue( x ){
  for(int i = rear+1; i<=length; i++){
    q[i] = q[i+1]; }
  }
```