

## 자료 구조 Lab 010 :

lab010.zip 파일 : LabTest.java lab010.java lab.in lab.out lab18010.pdf

-----

### 제출

lab010.java 를 **lab010\_학번.java** 로 변경하여 이 파일 한 개만 제출할 것.

---

다음은 사용자로부터 정렬이 된 숫자의 리스트 2개를 입력 받아 이를 합병 (Merge) 하는 과정을 구현하는 과제이다. Natural Merge Sort의 간단한 예이다.

수행 예는 다음과 같다.

```
sanghwan@PC: ~/dbox/classes201/ds/lab20/lab20010
sanghwan@PC-:~/dbox/classes201/ds/lab20/lab20010$ java LabTest
Merge > merge 1 3 5 7 9 2 4 6 8 -1
noe = 9, R = 5
L R L R L R L R
Input : 1 3 5 7 9 2 4 6 8
Output : 1 2 3 4 5 6 7 8 9

Merge > merge 4 1 2 3 -1
noe = 4, R = 1
R R R
Input : 4 1 2 3
Output : 1 2 3 4

Merge > merge 2 3 4 1 -1
noe = 4, R = 3
R
Input : 2 3 4 1
Output : 1 2 3 4

Merge > merge 2 3 7 1 4 5 8 9 -1
noe = 8, R = 3
R L L R R L
Input : 2 3 7 1 4 5 8 9
Output : 1 2 3 4 5 7 8 9

Merge > _
```

사용자가 사용하는 명령어의 syntax는 다음과 같다. main() 함수에 정의되어 있다.

- merge <number list>

사용자가 숫자를 입력하고 맨 마지막 숫자는 -1로서 입력의 끝을 표시하는데, 이 때 숫자들은 정렬이 된 2개의 리스트로 이루어져 있다. 즉 숫자가 증가하는 순서로 입력

이 되다가 한 번 감소하고 다시 증가하는 순서로 진행되어야 한다. 예를 들면 위 예제의 첫번째 목록은 (1 3 5 7 9) 와 (2 4 6 8)로 구분되는 것이다. 이 두 개의 리스트를 병합하여, 정렬된 하나의 리스트를 Output으로 보여준다.

위 예제에서 noe는 전체 숫자의 개수이고, R 은 두 번째 리스트가 시작하는 index 이다.

이 내용을 구현하기 위해 다음 하나의 함수를 구현해야 한다.

- `void Merge();`

`class NaturalMerge`에는 다음 3개의 변수가 있다.

```
int noe; // the number of elements
private int[] inputArray; // input array
int[] outputArray; // output array
```

이중 noe는 원소의 개수를 의미하고, `inputArray`는 입력된 숫자들이 들어 있는 array를 의미하며, `outputArray`는 정렬된 숫자들을 저장할 array 이다. `inputArray` 배열의 두 리스트를 병합하여 결과를 `outputArray [0]`부터 저장을 시작하면 된다.

이 함수를 구현하기 위해서는 우선 두번째 리스트가 시작하는 인덱스 R을 찾아내고, noe와 R을 출력한 후, 이 두 리스트를 병합하면 된다.

병합하는 알고리즘은 수업시간에 설명했던 merge sort의 병합 알고리즘을 사용하면 되고, 왼쪽 리스트와 오른쪽 리스트의 원소를 비교하여 `outputArray`에 추가할 때 이 추가되는 원소가 들어 있는 리스트를 L과 R 중 하나로 표시한다. L은 왼쪽 리스트를 의미하고, R은 오른쪽 리스트를 의미한다. 한쪽 리스트가 끝나면 더 이상 표시하지 않아도 된다. 단, 왼쪽 리스트의 원소와 오른쪽 리스트의 원소가 같을 경우는 왼쪽 리스트의 원소를 먼저 `outputArray`에 추가한다.