

답안은 메모장에 작성

채점시 이클립스에서 실행 -> 몇 번을 고쳐야 성공수행하느냐에 따라 양심적으로 채점하심이

만일 한 문제도 못 풀면 심각

## 문제 1) 버블 소트, 머지

문제 설명

A반의 국어, 영어, 수학 점수는 다음과 같다. (프로그램 작성시 점수는 임의로 넣어도 됨)

학번	이름	국어	영어	수학
101	홍길일	67	11	44
102	홍길이	55	77	33
103	홍길삼	77	66	99
104	홍길사	33	77	100
105	홍길오	46	34	23
106	홍길육	35	99	44
107	홍길칠	77	100	22
108	홍길팔	34	23	23
109	홍길구	99	44	66
110	홍길십	100	22	100

B반의 국어, 영어, 수학 점수는 다음과 같다.

학번	이름	국어	영어	수학
111	홍동일	11	44	44
112	홍동이	77	33	33
113	홍동삼	22	99	22
114	홍동사	23	100	23
115	홍동오	66	23	66
116	홍동육	100	44	100
117	홍동칠	100	22	22
118	홍동팔	23	23	23
119	홍동구	44	66	66
120	홍동십	22	100	100

## 문제

- 1) 각반에서 학생별 국영수 총점을 구한 후 오름차순으로 정렬하여 출력하시오 (단 소트는 버블소트, 정 모르면 아는 소트로 하시오)

### A반 총점 순위

학번	이름	총점
106	홍길육	300
109	홍길구	299
101	홍길일	298
~~~		

### B반 총점 순위

학번	이름	총점
116	홍동육	300
119	홍동구	299
111	홍동일	298
~~~		

- 2) A반과 B반을 병합하여 총점을 오름차순으로 정렬하여 출력하시오 (단 A,B반 총점이 오름차순으로 정렬이 되어 있는 데이터를 이용하여 병합)

### A,B반 총점 순위

학번	이름	총점
116	홍동육	300
109	홍길구	299
111	홍동일	298
~~~		

## 문제 2) 재귀호출 -기초

문제설명

통계학에서 조합(combination)은

서로 다른  $n$ 개에서  $r$ 개를 선택하는 방법의 수는

$${}_nC_r = \frac{{}_nP_r}{r!} = \frac{n!}{r! \cdot (n-r)!}$$

이고, 이것을 '조합'이라고 한다.

이다

문제

조합을 구하는 `public static int calculateCombination(int n, int r)` 를 작성하시오.

단 factorial함수(!)를 만들고 이함수는 recursive call(재귀적호출)을 사용하시오

### 문제3)이진검색(binary search)

문제설명

국어점수는 0점에서 100점 사이인데, a={1,2,10,12.....,100}을 랜덤함수를 통하여 1000명을 생성하였다.

```
import java.util.Arrays;
public class test3 {
    public static void main(String[] args) {
        // TODO Auto-generated method stub
        int a[]= new int[1001];
        for(int i=0;i<a.length-1;i++) {
            a[i]=(int) (Math.random()*100+1);
        }
        a[1001]=1000; //일단 1001은 밀어서 없어질놈으로 이렇게 처리
        Arrays.sort(a);
        for(int i=0;i<a.length-1;i++) { //1000개출력
            System.out.printf("%d)%d\n",i,a[i]);
        }
        ///////////////////////////////////
        // a어레이에 몇번째 항목에 88점이 삽입
        // 88점 위치를 이진검색으로 찾아 삽입후 미시오
        ///////////////////////////////////
        for(int i=0;i<a.length;i++) { //1001개출력
            System.out.printf("%d)%d\n",i,a[i]);
        }
    }
}
```


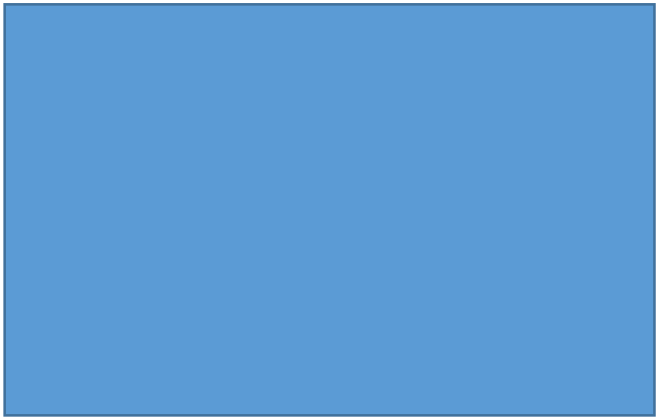

문제)

이때 88점이 누락됨을 알아서 a배열에 삽입하여야 한다 이 프로그램을 작성하시오.(단 이진검색으로 위치를 찾으시오)

모든 과정을 코딩하시오

### 문제3)DFS/BFS탐색

```
import java.util.*;
public class Test2 {
    private int V; // 정점의 개수
    private LinkedList<Integer>[] adj; // 인접 리스트

    public Test2(int v) {
        V = v;
        adj = new LinkedList[V];
        for (int i = 0; i < v; ++i)
            adj[i] = new LinkedList();
    }
    // 그래프에 간선 추가
    void addEdge(int v, int w) {
        adj[v].add(w);
    }
    // DFS 탐색
    void DFS(int v) {
        
    }
    void DFSUtil(int v, boolean[] visited) {
        
    }
    // BFS 탐색
    void BFS(int v) {
        
```



```
}
```

```
public static void main(String[] args) {
```

```
    Test2 g = new Test2(4);
```

```
    g.addEdge(0, 1);
```

```
    g.addEdge(0, 2);
```

```
    g.addEdge(1, 2);
```

```
    g.addEdge(2, 0);
```

```
    g.addEdge(2, 3);
```

```
    g.addEdge(3, 3);
```

```
    System.out.println("DFS 탐색 결과:");
```

```
    g.DFS(2);
```

```
    System.out.println("\nBFS 탐색 결과:");
```

```
    g.BFS(2);
```

```
}
```

```
}
```

문제 : 가려진 부분을 코딩하시오