

Merge Sort (Iterative)

재귀적이 아닌 반복문으로 Merge sort 프로그램을 구현하시오. Merge sort 를 반복문으로 구현하는 과정(강의노트 참조)에서 다음과 같이 정렬된 두 개의 부분배열을 merge 하는 과정(강의노트 참조)이 필요하다.

```
void merge(int a[], int low, int mid, int high)
{
    int i, j, k;
    int tmp[MAX_SIZE];

    for(i=low; i<=high; i++)
        tmp[i] = a[i];

    i = k = low;
    j = mid+1;
    while(i<=mid && j<=high)
        if(tmp[i] <= tmp[j])          // <---- 핵심연산자(비교연산자)
            a[k++] = tmp[i++];
        else
            a[k++] = tmp[j++];

    while(i<=mid)
        a[k++] = tmp[i++];
    while(j<=high)
        a[k++] = tmp[j++];
}
```

주어진 입력 데이터를 merge sort 를 실행하는 과정에서 위의 핵심연산자(비교연산자)의 실행 횟수를 계산하는 merge sort 알고리즘을 반복문으로 프로그램을 작성하시오.

입력

입력은 표준입력(standard input)을 사용한다. 입력은 t 개의 테스트 케이스로 주어진다. 입력의 첫 번째 줄에 테스트 케이스의 개수를 나타내는 정수 t 가 주어진다. 두 번째 줄부터 t 개의 줄에는 한 줄에 한 개의 테스트 케이스에 해당하는 데이터가 입력된다. 각 줄에서 첫 번째로 입력되는 정수 n ($1 \leq n \leq 100$)은 정렬하여야 할 정수의 개수를 나타낸다. 그 다음으로는 n 개의 정수가 입력된다. 이 정수는 최소 1이며 최대 10,000 이다. 각 정수들 사이에는 한 개의 공백이 있으며, 잘못된 데이터가 입력되는 경우는 없다.

출력

출력은 표준출력(standard output)을 사용한다. 입력되는 테스트 케이스의 순서대로 다음 줄에

이어서 각 테스트 케이스의 결과를 출력한다. 각 테스트 케이스에 해당하는 출력의 첫 번째 줄에 입력되는 정수를 merge sort 알고리즘으로 정렬하였을 때의 핵심 연산자의 실행 횟수를 출력한다.

입력과 출력의 예

입력	출력
3	12
8 1 2 3 4 5 6 7 8	16
8 8 1 7 2 6 3 5 4	0
1 5	