버블 정렬(Bubble Sort)

강의 노트에 제시된 버블 정렬(bubble sort) 알고리즘, 개선된 버블 정렬 알고리즘 1, 2를 각각 구현하고, 각 알고리즘에서 실행한 비교 연산자의 총 횟수와 비교 연산의 결과로 비교한 두 데이터를 교환(swap)한 횟수를 계산하는 프로그램을 작성하시오.

입력

입력은 표준입력(standard input)을 사용한다. 입력은 t 개의 테스트 케이스로 주어진다. 입력의 첫 번째 줄에 테스트 케이스의 개수를 나타내는 정수 t가 주어진다. 두 번째 줄부터 t 개의 줄에는 한 줄에 한 개의 테스트 케이스에 해당하는 데이터가 입력된다. 각 줄에서 첫 번째로 입력되는 정수 $n\,(1 \le n \le 1{,}000)$은 정렬하여야 할 정수의 개수를 나타낸다. 그 다음으로는 n개의 정수가 입력된다. 이 정수들은 최소 1이며 최대 10,000이다. 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. 먼저 버블 정렬 알고리즘에서 실행한 비교 연산자의 총 횟수와 교환 횟수를 출력하고, 그 다음으로는 개선 알고리즘 1, 그리고 마지막으로 개선 알고리즘 2의 비교 연산자의 총 횟수와 교환 횟수를 출력한다. 모두 6개 정수들 사이에는 한 개의 공백을 둔다.

입력과 출력의 예

입력	출력
3 9 1 2 3 4 5 6 7 8 9 9 9 8 7 6 5 4 3 2 1 9 9 6 3 1 2 4 5 7 8	36 0 8 0 8 0 36 36 36 36 36 36 36 15 26 15 20 15

BubbleSort.c

```c
#include <stdio.h>

#define MAX_SIZE 1000
void bubbleSort(int a[], int n);
void bubbleSortImproved1(int a[], int n);
void bubbleSortImproved2(int a[], int n);
void copyArray(int a[], int b[], int n);

int main()
{
    int numTestCases;

    scanf("%d", &numTestCases);
    for (int i = 0; i < numTestCases; ++i)
    {
        int num;
        int a[MAX_SIZE], b[MAX_SIZE];

        scanf("%d", &num);
        for (int j = 0; j < num; j++)
            scanf("%d", &b[j]);

        copyArray(a, b, num);
        bubbleSort(a, num);

        copyArray(a, b, num);
        bubbleSortImproved1(a, num);

        copyArray(a, b, num);
        bubbleSortImproved2(a, num);
        printf("\n");
    }

    return 0;
}

void swap(int* a, int* b)
{
    int tmp = *a;
    *a = *b;
    *b = tmp;
}

/* BubbleSort 함수 */
void bubbleSort(int a[], int n)
{
    int countCmpOps = 0;  // 비교 연산자 실행 횟수
    int countSwaps = 0;   // swap 함수 실행 횟수

    // bubble sort 알고리즘 구현

    printf("%d %d ", countCmpOps, countSwaps);
}
```

```
/* BubbleSort 함수 - Improved Version 1 */
void bubbleSortImproved1(int a[], int n)
{
    int countCmpOps = 0;  // 비교 연산자 실행 횟수
    int countSwaps = 0;   // swap 함수 실행 횟수

    // bubble sort의 개선된 알고리즘 (1) 구현

    printf("%d %d ", countCmpOps, countSwaps);
}

/* BubbleSort 함수 - Improved Version 2 */
void bubbleSortImproved2(int a[], int n)
{
    int countCmpOps = 0;  // 비교 연산자 실행 횟수
    int countSwaps = 0;   // swap 함수 실행 횟수

    // bubble sort의 개선된 알고리즘 (2) 구현

    printf("%d %d ", countCmpOps, countSwaps);
}

void copyArray(int a[], int b[], int n)
{
    for (int i = 0; i < n; i++)
        a[i] = b[i];
}
```