Non-Recursive Merge Sort

정렬(sorting)은 주어진 여러 개의 자료들을 특정한 순서로 나열하는 것을 말하는 것으로서, 정렬 알고리즘에는 가장 단순한 버블 정렬(bubble sorting) 이외에도 다양한 종류의 정렬 알고리즘이 있다. 이들 알고리즘 중에서도 가장 빠르고 효율적인 알고리즘은 퀵 정렬(quick sorting)이라고 알려져 있다. 정렬은 어떤 문제를 컴퓨터를 이용해서 해결하는데 있어서는 필수적인 기능으로서, 이를 위하여 C 프로그래밍 언어에서도 퀵 정렬을 표준 라이버러리 함수로서 제공하고 있다. C 언어에서 제공하는 퀵 정렬함수는 qsort()로서, 이 함수의 프로토타잎은 'stdlib.h'에 정의되어 있으며, 이 함수의 매개변수(혹은 인수)는 다음과 같이 정의된다.

```
#include <stdlib.h>

void qsort( void *base, size_t nel, size_t width,
                  int (*compare)(const void *, const void *));
```

함수 qsort() 는 1 차원 배열에 저장된 어떠한 종류의 자료에 대해서도 정렬해 준다. 이를 위해서 qsort() 함수에 자료가 저장된 1 차원 배열의 시작주소, 배열에 저장된 자료의 개수, 자료 한 개가 차지하는 메모리의 크기, 두 자료의 크기를 비교하는 함수에 관한 정보를 전달해주어야 한다. 함수 qsort()의 각 매개변수는 다음과 같다.

1. base : 정렬될 자료가 저장된 1 차원 배열의 시작 주소
2. nel : 배열에 저장된 자료의 개수
3. width : 배열에 저장된 한 개의 자료가 차지하는 메모리의 크기 (바이트 수)
4. compare : 배열에 저장된 두 개의 자료의 크기를 비교하는 함수 포인터

비교 함수 compare()는 비교할 두 자료의 주소를 매개변수로 가지며, 두 자료의 크기를 비교하여 첫 번째 자료가 두 번째 자료보다 큰 경우에는 임의의 양의 정수를 리턴하고, 두 자료의 크기가 같은 경우에는 0 을 리턴하며, 첫 번째 자료가 두 번째 자료보다 작은 경우에는 임의의 음의 정수를 리턴하는 함수이어야 한다. 또한 비교 함수 compare()는 범용 포인터(generic pointer)인 const void * 타잎(데이터 형)을 매개변수로 받도록 만들어야 한다.

다음은 1 차원 배열에 저장된 정수 값들을 오름차순으로 정렬하여 출력하는 프로그램의 예이다.

```
#include <stdio.h>
#include <stdlib.h>

int icompare(const void *a, const void *b)
{
        return *(int *)a - *(int *)b;
}
```

```
void main()
{
        int i;
        int ints[]={6, 9, 5, 1, 3, 2, 4, 8, 7};

        qsort(ints, sizeof(ints)/sizeof(ints[0]), sizeof(int), icompare);
//      msort(ints, sizeof(ints)/sizeof(ints[0]), sizeof(int), icompare);

        for (i=0; i<sizeof(ints)/sizeof(ints[0]); i++)
        {
                printf("%d\n", ints[i]);
        }
}
```

위에서 설명한 qsort()와 같은 매개변수를 가지는 non-recursive merge sort 인 msort() 프로그램을 작성하고, msort()를 이용하여 오름차순으로 정렬하는 프로그램을 작성하시오.

입력

입력 파일의 이름은 “input.txt” 이다. 입력은 t 개의 테스트 케이스로 주어진다. 입력 파일의 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 ($1 \le n \le 100$) 은 정렬하여야 할 정수의 개수를 나타낸다. 그 다음으로는 n 개의 정수가 입력된다. 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 줄에 입력되는 정수를 오름차순으로 출력한다. 각 정수들 사이에는 한 개의 공백을 둔다.

입력과 출력의 예

입력	출력
3 9 99 81 700 6 5005 444 321 23456 1 9 1 2 3 4 5 6 7 8 9 1 10000	1 6 81 99 321 444 700 5005 23456 1 2 3 4 5 6 7 8 9 10000