정렬(insertion, merge)

Lab(insertionSort)

- 임의의 정수 n을 입력 받아 n개의 정수 (10만 이하)의 난수를 만들어 이를 정렬하는 프로그램을 작성하시오. 정렬하는 부분은 함수로. **(insertion sort)**
  - 동적 할당 : n개의 정수를 저장할 수 있는 배열 생성
  - 배열에는 랜덤 정수를 저장
  - insertion sort로 정렬.
  - 프로그램 종료 전에 배열 반환

PREP(mergeSort를 위한 partition함수)

mergeSort를 프로그래밍하기 이전에 그 안에서 사용되는 merge함수를 정의하여 테스트 하려한다.

10, 20, 30, 70, 80, 11, 33, 44, 55, 66의 원소들을 갖는 list 배열은 0-4까지 정렬, 5-9까지 정렬되어 있다고 할 때, merge(list, 0, 4, 9)와 같이 호출하면 배열 list[0..4]와 list[5..9]를 합하여 정렬된 하나의 배열 list[0..9]가 만들어진다.

어떤 식으로 merge 함수를 정의할 것인가를 구상해보자.

```
#include <stdio.h>
#include <stdlib.h>
void merge(int A[], int p, int q, int r)
{
// 정렬되어 있는 두 배열 A[p ... q]와
// A[q+1 ... r]을 합하여
// 정렬된 하나의 배열 A[p .. r]을 만든다.
}

void printList(int A[], int s, int e)
{
    int i;
    for (i = s; i <= e; i++)
        printf("%d ", A[i]);
    printf("₩n");
}
```

```
int main(void)
{
    int list1[] =
        10, 20, 30, 70, 80,   // 0-4까지 정렬
        11, 33, 44, 55, 66};  // 5-9까지 정렬
    int list2[] =
        {10, 20, 30, 70, 80,  // 0-4까지 정렬
        11, 33, 44, 55, 66};  // 5-9까지 정렬

    // test #1
    printList(list1, 0, 9);
    merge(list1, 0, 4, 9);
    printList(list1, 0, 9);

    // test #2
    printList(list2, 2, 7);
    merge(list2, 2, 4, 7);
    printList(list2, 2, 7);
}
```

Lab(mergeSort)

- 임의의 정수 n을 입력 받아 n개의 정수 (10만 이하)의 난수를 만들어 이를 정렬하는 프로그램을 작성하시오. 정렬하는 부분은 함수로. **(merge sort)**
  - 동적 할당 : n개의 정수를 저장할 수 있는 배열 생성
  - 배열에는 랜덤 정수를 저장
  - merge sort로 정렬.
  - 프로그램 종료 전에 배열 반환