

실습과제 06

- 키보드로 부터 연속해서 정수들을 입력받는다. 정수가 하나 씩 입력될 때 마다 현재까지 입력된 정수들을 오름차순으로 정렬하여 화면에 출력한다. 단, 새로 입력된 정수가 이미 배열에 저장되어 있다면 추가하는 대신 “duplicate entry”라고 출력한다. 사용자가 -1을 입력하면 프로그램을 종료한다.

| 입력 예 | 출력 |
|------|-----------------|
| 5 | 5 |
| 2 | 2 5 |
| 5 | duplicate entry |
| 1 | 1 2 5 |
| 3 | 1 2 3 5 |
| -1 | |

- 두 개의 입력 파일 input1.txt와 input2.txt에 각각 정수들이 이미 오름차순으로 정렬된 상태로 저장되어 있다. 먼저 input1.txt 파일에 있는 정수들을 읽어 배열 data1에 순서대로 저장하고, input2.txt에 있는 정수들을 읽어 배열 data2에 순서대로 저장한다. 두 배열에 저장된 정수들을 배열 data3로 합쳐서 하나의 정렬된 배열을 만들어 출력하라. 두 파일에 저장된 정수들은 각각 최대 1000개 이다.

```
int main() {
    int data1[1000], data2[1000], data3[2000];
    int n1 = 0, n2 = 0, n3 = 0;
    FILE *fp1 = fopen("input1.txt", "r");
    FILE *fp2 = fopen("input2.txt", "r");
    while (!feof(fp1)) fscanf(fp1, "%d", &data1[n1++]);
    while (!feof(fp2)) fscanf(fp2, "%d", &data2[n2++]);
    fclose(fp1); fclose(fp2);

    /* 배열 data1과 data2에 오름차순으로 정렬되어 저장된 모든 정수들을 */
    /* 배열 data3에 오름차순으로 정렬이 되도록 저장하라. */

    for (int i=0; i<n3; i++) /* n3 = n1 + n2 */
        printf("%d\n", data3[i]);
}
```

| INPUT1.TXT | INPUT1.TXT | 출력 |
|---------------------|------------|-------------------------------|
| 2 4 5 8 10 11 13 40 | 3 9 11 28 | 2 3 4 5 8 9 10 11 11 13 28 40 |

- 선택정렬(selection sort) 알고리즘은 다음과 같다. 배열 data에 data[0]에서 data[n-1]까지 n개의 정수가 저장되어 있다. 먼저 data[0] ~ data[n-1] 사이의 정수들 중에서 가장 큰 정수를 찾는다. 그것을 data[k]라고 가정해보자. 그러면 data[k]와 data[n-1]을 swap한다. 이제 가장 큰 정수가 data[n-1], 즉 맨 마지막 위치에 저장되었으므로 그 값에 대해서는 더 이상 생각할 필요가 없다. 이제 data[0] ~ data[n-2] 중에서 최대값을 찾는다. 그 값을 data[p] 라고 하자. 그러면 다시 data[p] 와 data[n-2]를 swap하고 data[n-2]에 대해서는 잊어버려도 된다. 이런 식으로 계속하면 마지막에는 data[0]와 data[1] 중에 최대값을 data[1]과 swap하면 전체의 정렬이 완료된다. 이 알고리즘을 구현하라. 즉 input3.txt에 저장된 정수들을 읽어서 순서대로 배열에 저장한 후 selection sort 알고리즘으로 정렬하여 출력하는 프로그램을 작성하라.

4. 사용자로부터 n개의 정수를 입력받아 크기순으로 정렬한 후 중복된 수를 제거하고 출력하는 프로그램을 작성하라. 입력 형식은 먼저 n의 값이 주어지고 이어서 n개의 정수들이 주어진다. 예를 들어 n=8이고 입력된 정수들이 4 7 4 12 4 10 9 7이었다면 출력은 4 7 9 10 12이다. (추가: 실제 배열에서 중복된 값들을 모두 제거해야한다. 즉 배열에 값은 그대로 두고 출력할 때만 제외하고 출력하는 식으로 해서는 안된다.)

| 입력 예 | 출력 |
|---------------------|-------------|
| 8 4 7 4 12 4 10 9 7 | 4 7 9 10 12 |
| 5 1 1 1 1 1 | 1 |

5. 입력으로 n개의 구간(interval)이 주어진다. 각 구간은 구간의 시작점과 끝점으로 표현된다. 이 구간들을 시작점이 빠른 순서대로 정렬하여 출력하는 프로그램을 작성하라. 시작점이 같은 경우 끝점이 빠른 것을 먼저 출력한다. 입력 형식은 먼저 n의 값이 주어지고, 이어서 각 구간의 시작점과 끝점이 차례대로 주어진다. 각 구간의 시작점과 끝점은 정수이고, 끝점은 항상 시작점보다 크거나 같다.

| 입력 예 | 출력 |
|--------------------------|-------|
| 8 // 구간의 개수 | 1 5 |
| 2 8 // 첫 번째 구간의 시작점과 끝점 | 2 8 |
| 4 12 // 두 번째 구간의 시작점과 끝점 | 4 7 |
| 4 7 | 4 10 |
| 4 10 | 4 12 |
| 1 5 | 6 8 |
| 9 11 | 9 11 |
| 15 18 | 15 18 |
| 6 8 // 마지막 구간의 시작점과 끝점 | |

6. 입력으로 두 개의 양의 정수를 받는다. 그런 다음 두 정수를 사전식 순서(lexicographic order)로 출력하는 프로그램을 작성하라. 예를 들어 두 정수가 75와 111이라면 75는 7로 시작하므로 사전식 순서에서는 1로 시작하는 111보다 나중에 나와야 한다. 즉 111, 75의 순서로 출력해야 한다.

| 입력 예 | 출력 |
|------------|------------|
| 75 111 | 111 75 |
| 23441 2345 | 23441 2345 |
| 112 1111 | 1111 112 |
| 1231 123 | 123 1231 |
| 7333 9 | 7333 9 |

7. 입력으로 양의 정수들이 주어진다. 각 정수들은 8자리 이내의 정수들이다. 이 정수들을 사전식 순서(lexicographic order)로 정렬하여 출력하는 프로그램을 작성하라. 예를 들어 입력 정수들이 75, 111, 120, 3456, 9, 12, 67, 91이라면 111, 12, 120, 3456, 67, 75, 9, 91의 순서이다. 입력은 input6.txt 파일에서 읽는다. 정수의 개수는 따로 주어지지 않고 파일의 끝까지 읽어야 한다.

| 입력 예 | 출력 |
|---------------------------------|---------------------------------|
| 75 111 120 3456 9 12 67 91 | 111 12 120 3456 67 75 9 91 |
| 3467 3 78 3274 374 34213 123456 | 123456 3 3274 34213 3467 374 78 |