7. Selection Sort

Program Name: Selection.java **Input File: selection.dat**

The algorithm for a standard selection sort is:

1. Find the minimum value in the list
2. Swap it with the value in the first position
3. Repeat the steps above for the remainder of the list (starting at the second position the second time through the list, and advancing the starting position each time through the list).

Dr. Martin wants to trace what is happening with his data when he uses this selection sort to sort a non-empty array of integers. You are to write a program for him that will print his data after each iteration of the selection sort.

Input

The first line of input will contain a single integer `n` that indicates the number of integer arrays to follow. Each of the following `n` lines will contain the integers contained in an array. The integers in a given array will be separated by a single space.

Output

For each array, you will print the state of the array on one line after each iteration of the standard selection sort, printing a space after each array element. Print a blank line after the last iteration through the array.

Example Input File

```
3
45 15 12 -5 14 -3 8
12 -23 43 56 43 -1 -5
8 -2 -4 0 17 3 6 9 2 -2
```

Example Output to Screen

```
-5 15 12 45 14 -3 8
-5 -3 12 45 14 15 8
-5 -3 8 45 14 15 12
-5 -3 8 12 14 15 45
-5 -3 8 12 14 15 45
-5 -3 8 12 14 15 45

-23 12 43 56 43 -1 -5
-23 -5 43 56 43 -1 12
-23 -5 -1 56 43 43 12
-23 -5 -1 12 43 43 56
-23 -5 -1 12 43 43 56
-23 -5 -1 12 43 43 56

-4 -2 8 0 17 3 6 9 2 -2
-4 -2 8 0 17 3 6 9 2 -2
-4 -2 -2 0 17 3 6 9 2 8
-4 -2 -2 0 17 3 6 9 2 8
-4 -2 -2 0 2 3 6 9 17 8
-4 -2 -2 0 2 3 6 9 17 8
-4 -2 -2 0 2 3 6 9 17 8
-4 -2 -2 0 2 3 6 8 17 9
-4 -2 -2 0 2 3 6 8 9 17
```