

Section Handout 5

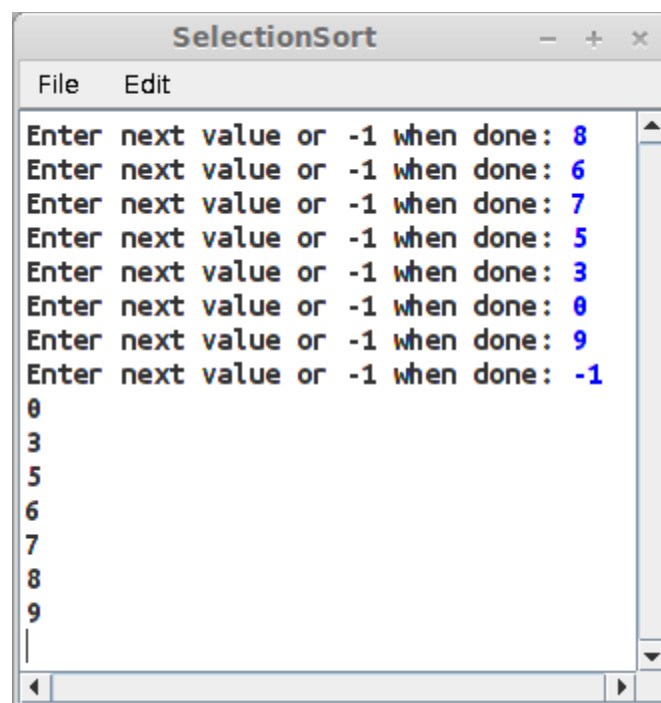
Problem One: Selection Sort

Write a program that reads a list of integers from the user, then prints them out in sorted order. The user should be able to enter values until they enter a sentinel value (say, -1), at which point the program prints the numbers back in sorted order.

To sort the numbers, you should use the *selection sort* algorithm, which works as follows:

- Find the smallest number in the list.
- Remove that number from the list and display it.
- Repeat until no numbers remain.

Here is a sample run of the program:



```
SelectionSort
File Edit
Enter next value or -1 when done: 8
Enter next value or -1 when done: 6
Enter next value or -1 when done: 7
Enter next value or -1 when done: 5
Enter next value or -1 when done: 3
Enter next value or -1 when done: 0
Enter next value or -1 when done: 9
Enter next value or -1 when done: -1
|
0
3
5
6
7
8
9
```

Problem Two: Method Testing

Below are a series of method signatures, along with a brief description of what the method should do. For each method, describe what specific tests you could run to confirm that the methods work correctly.

```
/**
 * Returns the largest value from a list of numbers. If the list is empty, returns
 * -1 as a sentinel.
 *
 * @param list The list in question.
 * @return The maximum value, or -1 if the list is empty.
 */
private int maxValue(ArrayList<Integer> list)

/**
 * Given a list of strings, returns whether all of those strings are distinct (i.e.
 * no two strings are the same).
 *
 * @param list The list in question.
 * @return Whether all of the strings are distinct.
 */
private boolean areAllDistinct(ArrayList<String> list)
```