

Homework – Topic 9 – Donations:

Write a complete program to do the following:

The main program calls a method to read in (from an input file) a set of people's three-digit ID numbers and their donations to a charity (hint: use parallel arrays). Then the main program calls a method to sort the ID numbers into numerical order, being sure to carry along the corresponding donations. The main program then calls a method to print the sorted lists in tabular form, giving both ID numbers and donations. Then the main program calls another method to sort the donation amounts into ascending order, carrying along the corresponding ID numbers. It then, once again, prints the sorted lists, giving both ID numbers and donations.

Here are the details:

(a) The main program calls a method to read in the data from a file. The data consists of sets of data, each of which contains a person's three-digit integer ID number and a donation in dollars and cents. (e.g., 456 200.00 or 123 302.34). The file is read until end-of-file is reached. The method returns how many sets of data were read in. The main program calls the return value donorCount.

The main program calls these arrays idNumbers and donations. A separate printing method prints the original set of data in the form of a neat table. When the arrays print, there should be an overall heading, plus headings for the columns of ID numbers and donations.

(b) Then the main program sends the array of ID numbers, the array of donations, and the size donorCount to a sorting method. This method sorts the ID numbers into numerical order using a **selectionSort**. Be sure to maintain the match-up of ID numbers and donations. For example, 456 should always be associated with 200.00, no matter where 456 moves in numerical order; similarly, 123 should stay with 302.34.

When the sorting method finishes and returns control to the main program, the main program calls the printing method to once again print the two arrays.

(c) Next, the main program sends the same three parameters to the second sorting method, which sorts the donations into numerical order (using a **bubbleSort**), being sure to maintain the linkup of ID numbers and donations.

When this sorting method finishes and returns control to the main program, the main program, once again, calls the printing method to print the two arrays with appropriate headings.

Your arrays should have room for up to 50 entries. To test the program, have a set of data with at least 15 to 20 values in each array. Make sure that your original order is not close to numerical order for either array and that the two numerical orders are not close to each other.

Required Submission:

1. The Java source code file (e.g., HW9.java)
2. The data input file of donors and donations(e.g., input.txt)
3. The program generated output file (e.g., output.txt)