

Background

In APP C16-1 EXT, you used the **bubble sort** algorithm to sort a list of integers. Bubble sort is just one of many different kinds of sorting algorithms, and it is actually very inefficient compared to others.

Selection sort is another sorting method. Like bubble sort, selection sort is a comparative method, but selection sort is faster and more efficient than bubble sort, though there are sorting methods that are much faster and more efficient than either of these.

Selection sort works by sorting an array one element at a time through selecting the minimum value amongst all the unsorted data and putting that in its proper spot in the sorted array. The sort starts with the first index of the array and steps through all indices. For each current index, the minimum value of the array from that current index to the end of the array is found. Then, that minimum value is swapped with the value at the current index, thus putting it in its proper place. The current index is then incremented, and a new minimum value of the array from the new current index to the end of the array is found and swapped into its proper place. This continues until the end of the array is reached and all minimum values have been moved to the front of the sorted array. You can see a cool GIF showing this process here: <https://en.wikipedia.org/wiki/File:Selection-Sort-Animation.gif>.

Problem Statement

A materials science company has collected data from various locations about experimentally determined thermal conductivities (in W/m*K) for different kinds of steel. The company has merged all the data into one data file (**APP_C27_1_EXT_steel.dat**), but the file still has text string labels for each individual location and the data is not in any order. There are 12 locations, and each location has exactly 20 reported thermal conductivities. You are to write a program to read in all the thermal conductivities while ignoring the location labels that will sort the data in ascending order using the selection sort method for future processing by the company.

Instructions

- Complete the C27-1 assignment which introduces simple file input in C.
- Copy the data file to your working directory. The file can be found here:
`/share/EED/class/engr1281/students/c/Class_27/Application/`
- Create a new program.
- Read all of the thermal conductivity data from the input file into a single array. You will have to ignore or throw out the location label text strings from the data file.
 - **Hint:** You might want to open the file and look at it to help determine the organization of the label text strings.
- Perform the selection sort method on the data array to sort the data into ascending order. The flowchart on the next page shows the steps of the selection sort method and can be used to write your own code.
- Print the sorted steel thermal conductivity data to the screen.
- Compare your code for the selection sort in this assignment to the code you wrote for the bubble sort in APP C16-1 EXT. What is one reason why selection sort is faster and more efficient than bubble sort? (There are two possible answers.) **Type your answer as a clearly labelled comment at the bottom of your code so it is easy to find and read (your answer can be as short as just one or two sentences).**

Selection Sort Flowchart

