Cooper Harris

Asm 6 Report

For asm6 I decided to use Russ' idea for a sorter, taking user input from the keyboard and using bubble sort to print out the array in ascending order. To use the program, call the function sortInput from a test file. It does not take any parameters. When you call this function a message asking how long you want the array to be will pop up in the run I/O tab at the bottom of the screen. Your answer should be typed as a single integer and will be used as the length of the array. The length must be between negative one and sixteen, if it is not a message will ask you to enter a valid length and will continue to do so until you enter a valid length. Next a message will ask for the numbers in the array. Type a single int and press enter, repeat this until you have typed as many integers as the length specified before. After you press enter after the last int a message at the bottom will print out the original array you entered and then print the sorted array right below it. A dialogue box will also pop up and ask if you want to run the program again. If you type Y, y, or click the ok button without putting in any text the program will run again repeating all the steps above. If you type N, n or click the cancel button the program will finish running.

The code is split up into several functions, one to manage each stage of the printing and sorting. The first function sortInput acts as the main class and does most of the work, calling other functions and setting up space on the stack for the user array. It calls the functions getLen, print_array, bubble_sort, and run_again to implement the other parts of the program. GetLen prints the message asking how long the array should be and takes an integer as input from the keyboard. Print_array does not take any user input and is only used to print an existing array in memory. Bubble_sort handles the sorting of a given array, it does this in place by using bubble sort to copy values from one index to the other to sort them in ascending order. Run_again uses syscall fifty-four to ask the user if they want to run the program again. It will call sortInput if the answer is yes.

The extra syscalls I used were: syscall five to get integer input from the keyboard, fifty-four which is an input dialogue string, and seventeen, the exit syscall. I used syscall fifty-four to create a dialogue box that asked the user if they wanted to run the program again. Syscall five was the most used because I had to call it every time I needed keyboard input. I originally wanted to use a dialogue box to get the integer input, but I decided that I liked how using the Run I/O textbox for this looked better.