* **Issues encountered sorting large arrays and vectors with recursion. How did you fix the issue(s) encountered.**

Sorting large arrays and vectors with recursion was taking more time and often hangs up the VISUAL STUDIO on sorting very large arrays.

So one way we can fix this issue by making sure that local temporary and function variables are declared as short int since recursive function calls are repeated and consume a lot of memory.

The best way to fix this issue is to use iterative approach.

* **The difference in performance between sorting with recursion version iteration. Which performed better, and why?**

Sorting with iteration performed better than sorting with recursion because:

1. Recursive approach involved overhead of repeated function calls. In this Selection sort example, It called the same function n times (n = array size).
2. Each method execution takes place in a stack. Recursion algorithms increases the stack size consuming a lot of memory. For selection sort, we are storing two integer variables to know the position and value of the smallest/largest number encountered so far. Hence, it creates an additional space of ( integer size \* number of array elements). This is not in the case of iteration since these values are only declared once and used for all steps during Selection sorting.
3. Hence, overhead of repeated function calls and extra memory is omitted in iterative approach.

* **The difference in performance between sorting C-Style arrays versus the STL vector. Which performed better, and why?**

Sorting C-Style arrays performed better than sorting STL vector because:

1. Vectors add extra instructions to check that you don't access beyond the end of array. This check can be time consuming since there were lot of such checks required during Slection sorting.