

## C++ Sorts Lab Assignments 11-1-2023

1.	<b>Quick Sort</b> : Code quick sort for i) ascending order ii) descending order
2.	<b>Selection Sort</b> <b>i)</b> code two functions minindex() , selectionsort() and call them in main() <b>ii)</b> code minindex() as recursive function and call it in selectionsort() function. Call the selectionsort() in main().
3.	<p>Given a circularly sorted integer array, code to find the total number of times the array is rotated. The rotation is in the anti-clockwise direction and there are no duplicates in the array.</p> <p>For example,  <b>Input:</b> nums = [7, 9, 10, 3, 4, 6]  <b>Output:</b> The array is rotated 3 times as the sorted array is [ 3, 4, 6, 7, 9, 10 ]</p> <p><b>Input:</b> nums = [3, 5, 7, 8, 9, 10]  <b>Output:</b> The array is rotated 0 times</p>
4.	<b>Cocktail Sort</b> Cocktail sort is also called as <b>bi-directional</b> bubble sort. 1. In first stage, similar to the bubble sort, loop through the array from left to right. The adjacent elements are compared, and if the left element is greater than the right one, we swap those elements. The largest element of the list is placed at the end of the array in the forward pass. 2. In second stage, loop through the array from the rightmost unsorted element to the left. The adjacent elements are compared, and if the right element is smaller than the left element then, we swap those elements. The smallest element of the list is placed at the beginning of the array in the backward pass. This process continues until the array elements are not sorted.
	<div style="display: flex; justify-content: space-between;"> <div style="width: 48%;"> <p>Logical function :</p> <pre> cocktailSort(a, n) beg = 0 , end = n-1 while( ....) { for i in range from beg to end { ..... ..... } end = end - 1 for i in range from end -1 to beg { if (a[i] &gt; a[i + 1]) ..... } start = start + 1 } </pre> </div> <div style="width: 48%; text-align: center;"> </div> </div>

5.	Code to print any one pattern of your favourite, using only do..while loops.
6.	<p><b>Ternary search</b> is a searching technique used to find out the position of any given element in a sorted array. While the array is divided into two parts for binary search where just one mid element is used, ternary search requires the array to be divided into three parts and has two mid elements.</p> <p>Logical Method :</p> <p>The key is compared with the first mid element, 'mid1'. If equal, 'mid1' is returned. If not equal, the key is next compared with 'mid2', and the same would be returned if equal.</p> <p>If not equal to either 'mid1' or 'mid2', the key is checked to be lesser than 'mid1'. If it is, then we recur to the first part.</p> <p>If it is not, then the key is checked to be greater than 'mid2'. If yes, then we will recur to the third part of the array.</p> <p>If not, then we recur to the middle part of the array.</p> <p>Finding mid1 and mid2</p> $\text{mid1} = \text{start} + (\text{end} - \text{start}) / 3;$ $\text{mid2} = \text{end} - (\text{end} - \text{start}) / 3;$
7.	<p><b>Insertion Sort :</b></p> <p>Inserting an element into a sorted subset. Initially, the sorted subset consists of only one first element at index 0. Then for each iteration, insertion sort removes the next element from the unsorted subset, finds the location it belongs within the sorted subset and inserts it there by moving the elements to right. It repeats until no input elements remain as shown below. Code to sort.</p> <p>Note the usage of for and while loops in the function.</p>
	<div> </div> <div> <p>Sample logical code.</p> <pre>void insertionsort(int arr[], int n) {     for ( )     {         .....         while ( )         {             .....         }         .....     } }</pre> </div>
<p style="text-align: center;">“Perhaps the most valuable result of an education is the ability to make yourself Do the thing you have to Do, when it ought to be done, whether you like it Or Not.”– Thomas Huxley</p> <div> <p><b>Aim at your Wills</b> <b>Sum up your Skills</b></p> <p style="text-align: center;"><b>Today's Struggle is</b> <b>Tomorrow's Strength</b> <b>That's Selection day's Success</b></p> <p style="text-align: right;">~KR</p> </div>	