

Week 12

Write a program that implements the following sorting methods to sort a

given list of integers in ascending order

i) SelectionSort ii) Quick Sort

i) SelectionSort

```
#include <stdio.h>

void swap(int *xp, int *yp)
{
    int temp = *xp;
    *xp = *yp;
    *yp = temp;
}

void selectionSort(int arr[], int n)
{
    int i, j, min_idx;
    // One by one move boundary of unsorted subarray
    for (i = 0; i < n-1; i++)
    {
        // Find the minimum element in unsorted array
        min_idx = i;
        for (j = i+1; j < n; j++)
            if (arr[j] < arr[min_idx])
                min_idx = j;
    }
}
```

```
// Swap the found minimum element with the first element
swap(&arr[min_idx], &arr[i]);
}
}

/* Function to print an array */
void printArray(int arr[], int size)
{
    int i;
    for (i=0; i < size; i++)
        printf("%d ", arr[i]);
    printf("\n");
}

// Driver program to test above functions
int main()
{
    int arr[] = {64, 25, 12, 22, 11};
    int n = sizeof(arr)/sizeof(arr[0]);
    selectionSort(arr, n);
    printf("Sorted array: \n");
    printArray(arr, n);
    return 0;
}
```

ii) Quick Sort

```
#include<stdio.h>
#include<conio.h>
void quickSort(int [10],int,int);
void main(){
int list[20],size,i;
printf("Enter size of the list: ");
scanf("%d",&size);
printf("Enter %d integer values: ",size);
for(i = 0; i < size; i++)
scanf("%d",&list[i]);
quickSort(list,0,size-1);
printf("List after sorting is: ");
for(i = 0; i < size; i++)
printf(" %d",list[i]);
getch();
}
void quickSort(int list[10],int first,int last){
int pivot,i,j,temp;
if(first < last){
pivot = first;
i = first;
j = last;
while(i < j){
```

```
while(list[i]<= list[pivot] && i < last)
    i++;
while(list[j] > list[pivot])
    j--;
if(i < j){
    temp = list[i];
    list[i] = list[j];
    list[j] = temp;
}
}
temp = list[pivot];
list[pivot] = list[j];
list[j] = temp;
quickSort(list,first,j-1);
quickSort(list,j+1,last);
}
}
```