

EXPERIMENT NO. 1

1) Selection Sort Algorithm

Program:-

```
#include <stdio.h>
```

```
void selectionSort(int arr[], int n) {  
    int i, j, minIndex, temp;  
    for (i = 0; i < n - 1; i++) {  
        minIndex = i;  
        for (j = i + 1; j < n; j++) {  
            if (arr[j] < arr[minIndex]) {  
                minIndex = j;  
            }  
        }  
        // Swap arr[i] and arr[minIndex]  
        temp = arr[i];  
        arr[i] = arr[minIndex];  
        arr[minIndex] = temp;  
    }  
}
```

```
int main() {  
    int arr[] = {64, 25, 12, 22, 11};  
    int n = sizeof(arr) / sizeof(arr[0]);  
    printf("Array before sorting:\n");  
    for (int i = 0; i < n; i++) {  
        printf("%d ", arr[i]);  
    }  
    printf("\n");  
  
    selectionSort(arr, n);  
  
    printf("Array after sorting:\n");  
    for (int i = 0; i < n; i++) {  
        printf("%d ", arr[i]);  
    }  
    printf("\n");  
  
    return 0;  
}
```

```
/tmp/DvBnUHyJ19.o
```

Array before sorting:

64 25 12 22 11

Array after sorting:

11 12 22 25 64

=== Code Execution Successful ===

2) Insertion Sort Algorithm:-

Program:-

```
#include <stdio.h>
```

```
void insertionSort(int arr[], int n) {
    int i, key, j;
    for (i = 1; i < n; i++) {
        key = arr[i];
        j = i - 1;

        /* Move elements of arr[0..i-1], that are greater than key,
           to one position ahead of their current position */
        while (j >= 0 && arr[j] > key) {
            arr[j + 1] = arr[j];
            j = j - 1;
        }
        arr[j + 1] = key;
    }
}
```

```
int main() {
    int arr[] = {64, 25, 12, 22, 11};
    int n = sizeof(arr) / sizeof(arr[0]);
    printf("Array before sorting:\n");
    for (int i = 0; i < n; i++) {
        printf("%d ", arr[i]);
    }
    printf("\n");

    insertionSort(arr, n);
```

```
    printf("Array after sorting:\n");  
    for (int i = 0; i < n; i++) {  
        printf("%d ", arr[i]);  
    }  
    printf("\n");  
  
    return 0;  
}
```

```
/tmp/WFA3wIctEq.o  
Array before sorting:  
64 25 12 22 11  
Array after sorting:  
11 12 22 25 64
```

```
=== Code Execution Successful ===
```