Name: Amal Thundiyil
Department: IT
Batch: D
UID: 2020400066

Aim: Implement Insertion and selection sort both. Show all intermediate passes and give analysis in terms of no of Comparisons and Exchanges

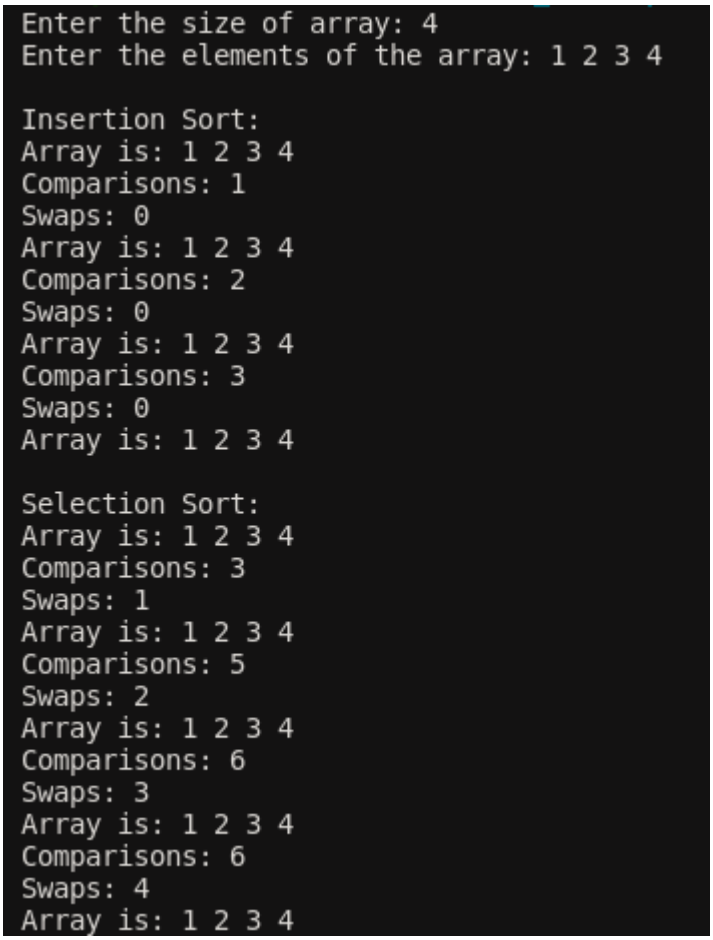
Program:

```
#include <stdio.h>

void display(int arr[], int n) {
   printf("Array is: ");
   for (int i = 0; i < n; i++) {
      printf("%d ", arr[i]);
   }
   printf("\n");
}

void insertion_sort(int arr[], int n) {
   int comparisons = 0, swaps = 0;
   for (int i = 1; i < n; i++) {
      int key = arr[i];
      int j = i - 1;
      while (j >= 0) {
         comparisons += 1;
         if (arr[j] > key) {
            arr[j + 1] = arr[j];
            j -= 1;
            swaps += 1;
         } else
            break;
      }
      arr[j + 1] = key;
      display(arr, n);
      printf("Comparisons: %d\n", comparisons);
      printf("Swaps: %d\n", swaps);
   }
}

void selection_sort(int arr[], int n) {
   int comparisons = 0, swaps = 0;
   for (int i = 0; i < n; i++) {
```

```
        int min_index = i;
        for (int j = i + 1; j < n; j++) {
            comparisons += 1;
            if (arr[min_index] > arr[j]) {
                min_index = j;
            }
        }
        swaps += 1;
        int temp = arr[i];
        arr[i] = arr[min_index];
        arr[min_index] = temp;
        display(arr, n);
        printf("Comparisons: %d\n", comparisons);
        printf("Swaps: %d\n", swaps);
    }
}

void main() {
    int n;
    printf("Enter the size of array: ");
    scanf("%d", &n);
    int arr1[n], arr2[n];
    printf("Enter the elements of the array: ");
    for (int i = 0; i < n; i++) {
        scanf("%d", &arr1[i]);
        arr2[i] = arr1[i];
    }
    printf("\nInsertion Sort: \n");
    insertion_sort(arr1, n);
    display(arr1, n);
    printf("\nSelection Sort: \n");
    selection_sort(arr2, n);
    display(arr2, n);
}
```

Output:

Best Case:



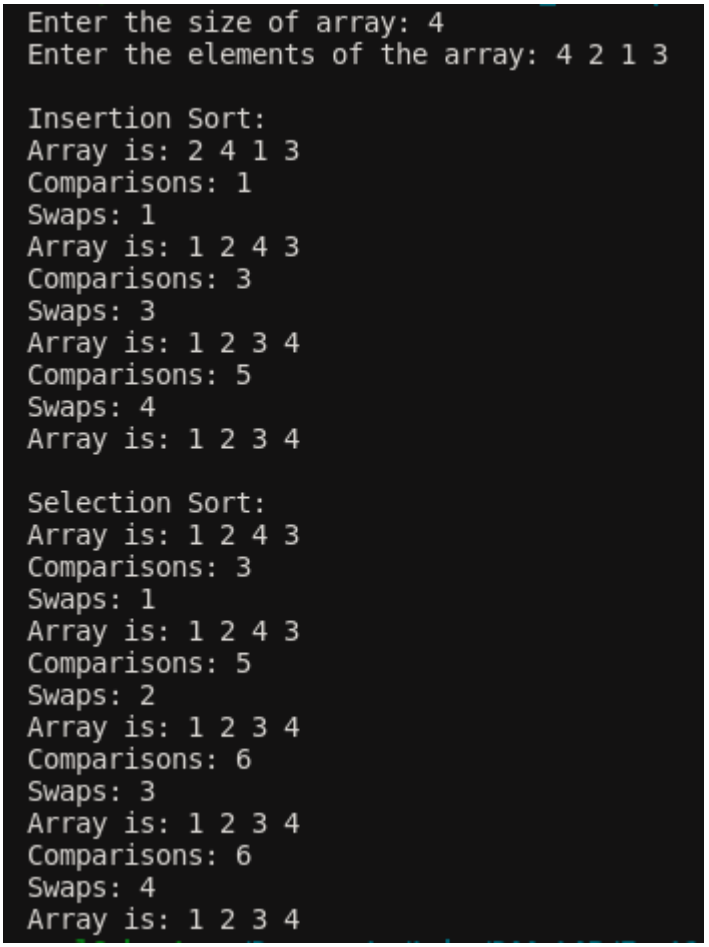
The number of swaps is 0 for insertion sort in the best case. The number of comparisons is 3. The number of swaps is 4 for selection sort in the best case. The number of comparisons is 6.

Worst Case:

```
Enter the size of array: 4
Enter the elements of the array: 4 3 2 1

Insertion Sort:
Array is: 3 4 2 1
Comparisons: 1
Swaps: 1
Array is: 2 3 4 1
Comparisons: 3
Swaps: 3
Array is: 1 2 3 4
Comparisons: 6
Swaps: 6
Array is: 1 2 3 4

Selection Sort:
Array is: 1 3 2 4
Comparisons: 3
Swaps: 1
Array is: 1 2 3 4
Comparisons: 5
Swaps: 2
Array is: 1 2 3 4
Comparisons: 6
Swaps: 3
Array is: 1 2 3 4
Comparisons: 6
Swaps: 4
Array is: 1 2 3 4
```

The number of swaps is 6 for insertion sort in the worst case. The number of comparisons is 6.
The number of swaps is 4 for selection sort in the worst case. The number of comparisons is 6.

Random Array Case:



The number of swaps is 4 for insertion sort in the random array case. The number of comparisons is 5.

The number of swaps is 4 for selection sort in the random array case. The number of comparisons is 6.