

# Digital and Analysis of Algorithm

---

## Lab 1

**Name: Harsh Brahmecha**

**PRN: 20220802003**

### 1. Write a C code to implement bubble sort algorithm

```
#include <stdio.h>

void bubble_sort(int arr[], int n) {
    int i, j;
    for (i = 0; i < n - 1; i++) {
        for (j = 0; j < n - i - 1; j++) {
            if (arr[j] > arr[j + 1]) {
                int temp = arr[j];
                arr[j] = arr[j + 1];
                arr[j + 1] = temp;
            }
        }
    }
}

int main() {
    int arr[] = {13, 32, 26, 35, 10,};
    int n = sizeof(arr) / sizeof(arr[0]);
    bubble_sort(arr, n);
    printf("Sorted array: ");
    for (int i = 0; i < n; i++) {
        printf("%d ", arr[i]);
    }
}
```

```
return 0;
}
```

Output:

```
harsh@Harshs-MacBook-Air Sem2 C % ./a.out
Sorted array: 10 13 26 32 35 %
```

## 2. Write a C code to implement insertion sort algorithm

```
#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;

        while (j >= 0 && arr[j] > key)
        {
            arr[j + 1] = arr[j];
            j = j - 1;
        }
        arr[j + 1] = key;
    }
}

void printArray(int arr[], int n)
{

```

```

    int i;
    for (i = 0; i < n; i++)
        printf("%d ", arr[i]);
    printf("\n");
}

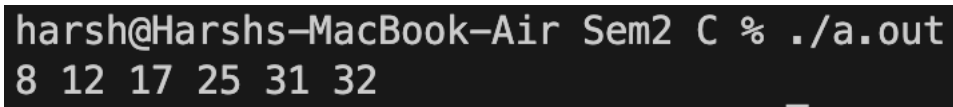
int main()
{
    int arr[] = {12, 31, 25, 8, 32, 17};
    int n = sizeof(arr) / sizeof(arr[0]);

    insertionSort(arr, n);
    printArray(arr, n);

    return 0;
}

```

Output:



```

harsh@Harshs-MacBook-Air Sem2 C % ./a.out
8 12 17 25 31 32

```

### 3. Write a C code to implement selection sort algorithm

```

#include <stdio.h>

void swap(int *a, int *b) {
    int temp = *a;
    *a = *b;
    *b = temp;
}

```

```

void selectionSort(int array[], int size) {
    for (int step = 0; step < size - 1; step++) {
        int min_idx = step;
        for (int i = step + 1; i < size; i++) {

            if (array[i] < array[min_idx])
                min_idx = i;
        }

        swap(&array[min_idx], &array[step]);
    }
}

void printArray(int array[], int size) {
    for (int i = 0; i < size; ++i) {
        printf("%d ", array[i]);
    }
    printf("\n");
}

int main() {
    int data[] = { 12, 29,25 ,8,32,17,40};
    int size = sizeof(data) / sizeof(data[0]);
    selectionSort(data, size);
    printf("Sorted array in Acsending Order:\n");
    printArray(data, size);
}

```

Output:

```

Sorted array in Acsending Order:
8 12 17 25 29 32 40

```