NAME :SAI LAKSHMI TEJASRI K

ROLL NO: CH.SC.U4CSE24243

1) BUBBLE SORT

Open

bubblesort.c
~/Downloads

```c
#include<stdio.h>
void bubblesort(int arr[], int n){
int i,j,temp;
for(i=0;i<n;i++){
   for(j=0;j<n-i-1;j++){
   if(arr[j] > arr[j+1]){
   temp=arr[j];
   arr[j]=arr[j+1];
   arr[j+1]=temp;
   }
   }
 }
}
void printarray(int arr[],int n){
int i;
for(i=0;i<n;i++){
   printf("%d ",arr[i]);
  }
  printf("\n");
}
int main(){
int arr[]={24,36,12,63,77,8,13};
int n=(sizeof (arr)/ sizeof arr[0]);
printarray(arr,n);
bubblesort(arr,n);
printf("the sorted array is\n");
printarray(arr,n);
return 0;
}
```

```
amma@amma39:~/Downloads$ gcc bubblesort.c -o bubblesort
amma@amma39:~/Downloads$ ./bubblesort
24 36 12 63 77 8 13
the sorted array is
8 12 13 24 36 63 77
```

2)INSERTION SORT

Open

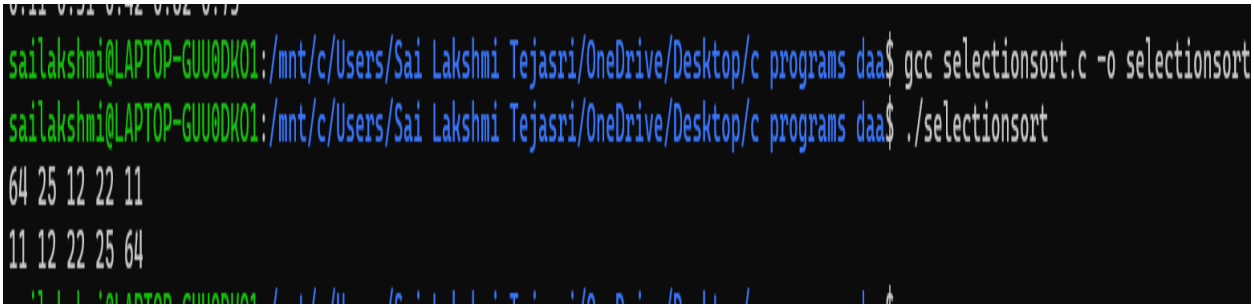
insertionsort.c
~/Downloads

```c
#include <stdio.h>
void insertionsort(int arr[], int n) {
    int i, j, key;
    for (i = 1; i < n; i++) {
        key = arr[i];
        j = i - 1;
        while (j >= 0 && arr[j] > key) {
            arr[j + 1] = arr[j];
            j--;
        }
        arr[j + 1] = key;
        }
}
void printarray(int arr[], int n) {
    for (int i = 0; i < n; i++)
        printf("%d ", arr[i]);
    printf("\n");
}
int main() {
    int arr[] = {12, 11, 13, 5, 6};
    int n = sizeof(arr) / sizeof(arr[0]);
    printarray(arr, n);
    insertionsort(arr, n);
    printf("elements after array is sorted\n");
    printarray(arr, n);
    return 0;
}
```

```
amma@amma39:~/Downloads$ gcc insertionsort.c -o insertionsort
amma@amma39:~/Downloads$ ./insertionsort
12 11 13 5 6
elements after array is sorted
5 6 11 12 13
```

3)SELECTION SORT

```c
#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;
            }
        }
        temp = arr[i];
        arr[i] = arr[minindex];
        arr[minindex] = temp;
    }
}
void printarray(int arr[], int n) {
    for (int i = 0; i < n; i++) {
        printf("%d ", arr[i]);
    }
    printf("\n");
}
int main() {
    int arr[] = {64, 25, 12, 22, 11};
    int n = sizeof(arr) / sizeof(arr[0]);
    printarray(arr, n);
    selectionsort(arr, n);
    printarray(arr, n);
    return 0;
}
```



4)BUCKET SORT

```c
#include <stdio.h>
#define MAX 20
#define BUCKETS 10
void insertionSort(float arr[], int size) {
    int i, j;
    float key;
    for (i = 1; i < size; i++) {
        key = arr[i];
        j = i - 1;
        while (j >= 0 && arr[j] > key) {
            arr[j + 1] = arr[j];
            j--;
        }
        arr[j + 1] = key;
    }
}
void bucketSort(float arr[], int n) {
    float buckets[BUCKETS][MAX];
    int bucketSize[BUCKETS] = {0};
    int i, j, k = 0;
    for (i = 0; i < n; i++) {
        int index = (int)(arr[i] * BUCKETS);
        if (index >= BUCKETS)
            index = BUCKETS - 1;
        buckets[index][bucketSize[index]++] = arr[i];
    }
    for (i = 0; i < BUCKETS; i++)
        insertionSort(buckets[i], bucketSize[i]);
    for (i = 0; i < BUCKETS; i++)
        for (j = 0; j < bucketSize[i]; j++)
            arr[k++] = buckets[i][j];
}
void printArray(float arr[], int n) {
    int i;
    for (i = 0; i < n; i++)
        printf("%.2f ", arr[i]);
    printf("\n");
}
```

```c
int main() {
    int n, i;
    float arr[MAX];
    printf("Enter number of elements: ");
    scanf("%d", &n);
    printf("Enter elements (0 to 1 range):\n");
    for (i = 0; i < n; i++)
        scanf("%f", &arr[i]);
    printf("Before sorting:\n");
    printArray(arr, n);
    bucketSort(arr, n);
    printf("After sorting:\n");
    printArray(arr, n);
    return 0;
}
```

```
sailakshmi@LAPTOP-GUU0DKO1:/mnt/c/Users/Sai Lakshmi Tejasri/OneDrive/Desktop/c programs daa$ gcc bucketsort.c -o bucketsort
sailakshmi@LAPTOP-GUU0DKO1:/mnt/c/Users/Sai Lakshmi Tejasri/OneDrive/Desktop/c programs daa$ ./bucketsort
Enter number of elements: 5
Enter elements (0 to 1 range):
0.42 0.31 0.52 0.73 0.11
Before sorting:
0.42 0.31 0.52 0.73 0.11
After sorting:
0.11 0.31 0.42 0.52 0.73
```

5) HEAP SORT

```c
#include <stdio.h>
void heapify(int a[], int n, int i) {
    int largest = i, l = 2*i + 1, r = 2*i + 2;
    if(l < n && a[l] > a[largest]) largest = l;
    if(r < n && a[r] > a[largest]) largest = r;
    if(largest != i) {
        int t = a[i];
        a[i] = a[largest];
        a[largest] = t;
        heapify(a, n, largest);}}
void heapSort(int a[], int n) {
    for(int i = n/2 - 1; i >= 0; i--)
        heapify(a, n, i);
    for(int i = n - 1; i > 0; i--) {
        int t = a[0];
        a[0] = a[i];
        a[i] = t;
        heapify(a, i, 0);}}
void printArray(int a[], int n) {
    for(int i = 0; i < n; i++)
        printf("%d ", a[i]);
    printf("\n");}
int main() {
    int a[] = {12, 11, 13, 5, 6, 7};
    int n = sizeof(a)/sizeof(a[0]);
    printf("Array before sorting:\n");
    printArray(a, n);
    heapSort(a, n);
    printf("Array after sorting:\n");
    printArray(a, n);
    return 0;}
```

```
Array before sorting:
12 11 13 5 6 7
Array after sorting:
5 6 7 11 12 13
```