AMRITA VISHWA VIDYAPEETHAM

DEEMED TO BE UNIVERSITY UNDER SECTION 3 OF UGC ACT, 1956

SCHOOL OF COMPUTING

Kishore S

CH.SC.U4CSE24222

Week – 2

Design and Analysis of Algorithm(23CSE211)

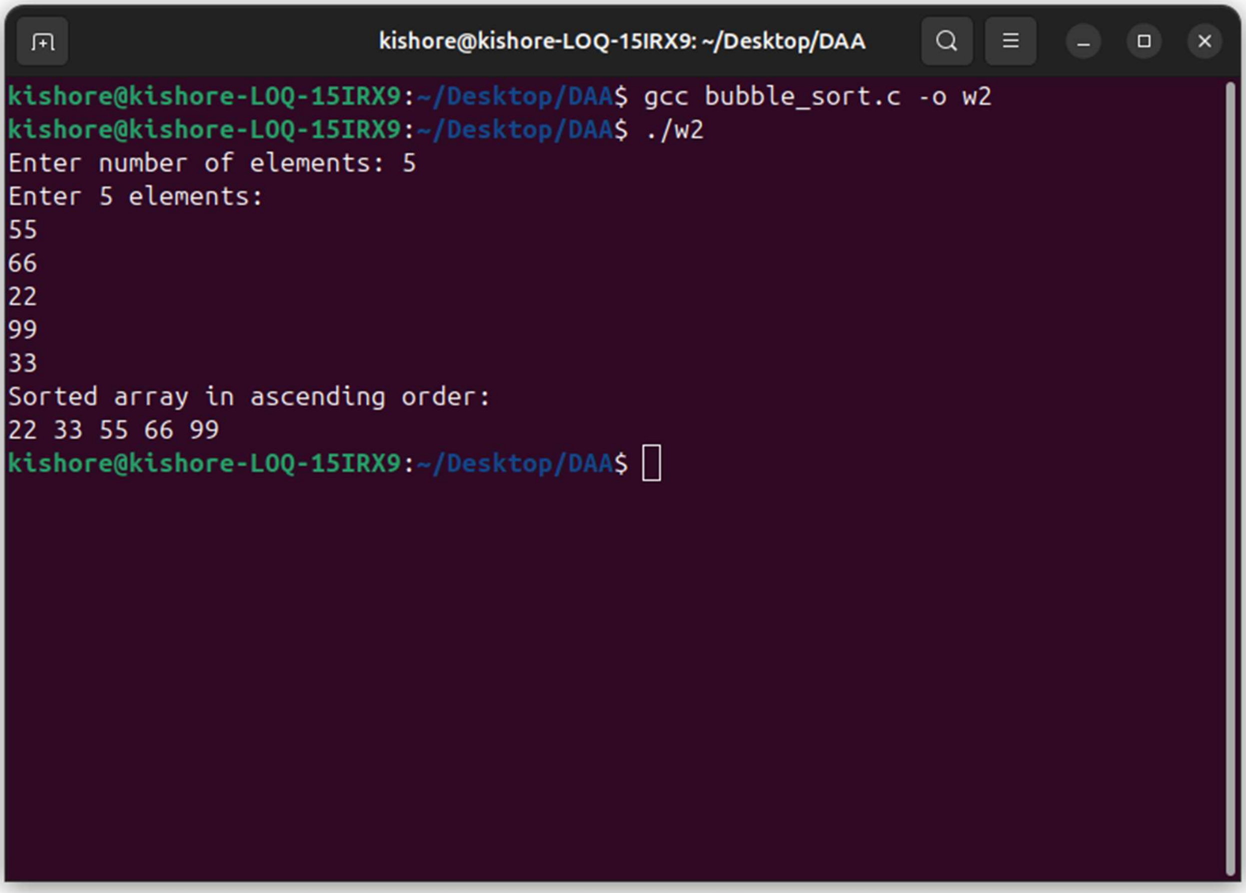
Sorting Techniques

1. Write a c program for bubble sort

Code:

```
#include <stdio.h>

int main() {
    int n, i, j, temp;
    int arr[100];

    printf("Enter number of elements: ");
    scanf("%d", &n);

    printf("Enter %d elements:\n", n);
    for (i = 0; i < n; i++) {
        scanf("%d", &arr[i]);
    }

    // Bubble Sort logic
    for (i = 0; i < n - 1; 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;
            }
        }
    }

    printf("Sorted array in ascending order:\n");
    for (i = 0; i < n; i++) {
        printf("%d ", arr[i]);
    }7
    printf("\n");

    return 0;
}
```

Output:



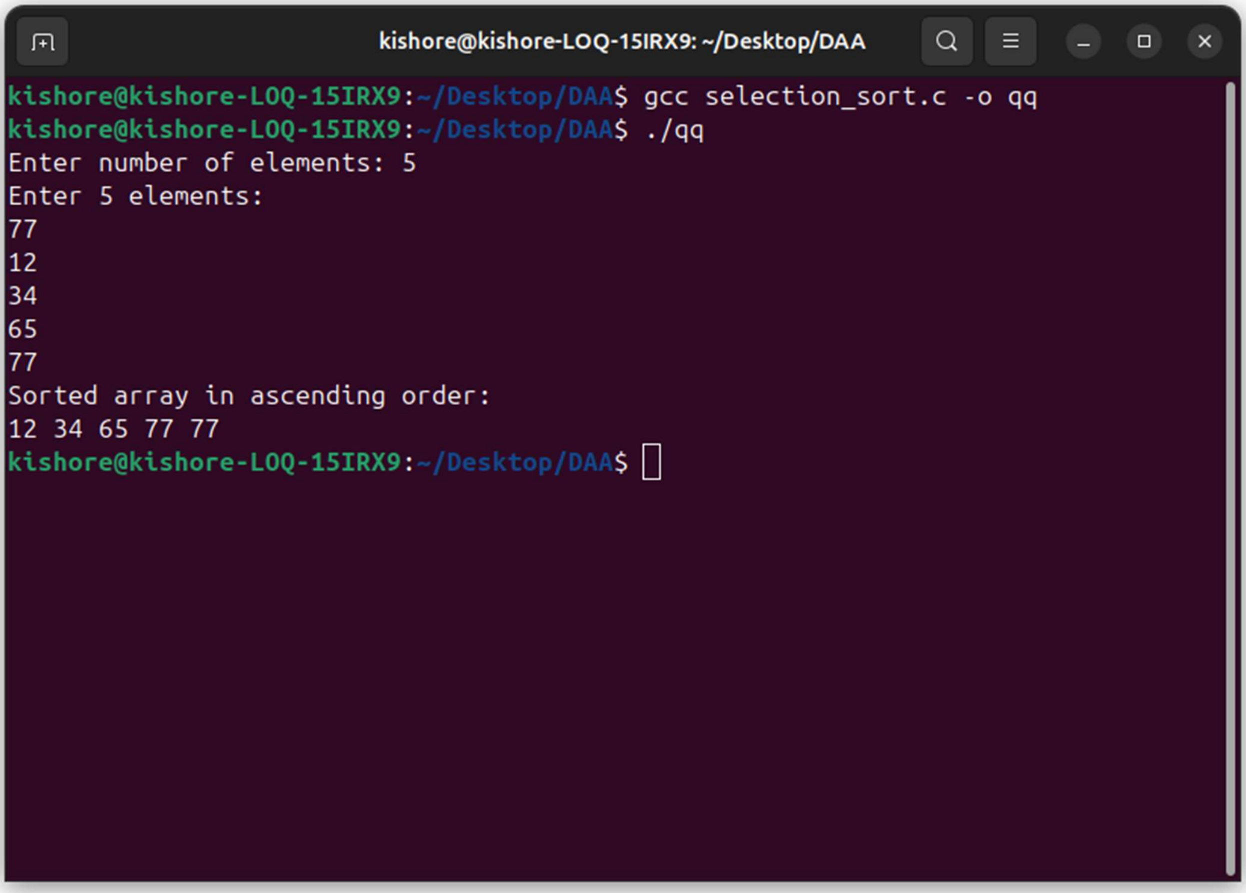
```
kishore@kishore-LOQ-15IRX9:~/Desktop/DAA$ gcc bubble_sort.c -o w2
kishore@kishore-LOQ-15IRX9:~/Desktop/DAA$ ./w2
Enter number of elements: 5
Enter 5 elements:
55
66
22
99
33
Sorted array in ascending order:
22 33 55 66 99
kishore@kishore-LOQ-15IRX9:~/Desktop/DAA$ 
```

2. Write a c program for selection sort

Code:

```
#include <stdio.h>

int main() {
    int n, i, j, minIndex, temp;
    int arr[100];

    printf("Enter number of elements: ");
    scanf("%d", &n);

    printf("Enter %d elements:\n", n);
    for (i = 0; i < n; i++) {
        scanf("%d", &arr[i]);
    }

    // Selection Sort logic
    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;
    }

    printf("Sorted array in ascending order:\n");
    for (i = 0; i < n; i++) {
        printf("%d ", arr[i]);
    }
    printf("\n");

    return 0;
}
```

Output:



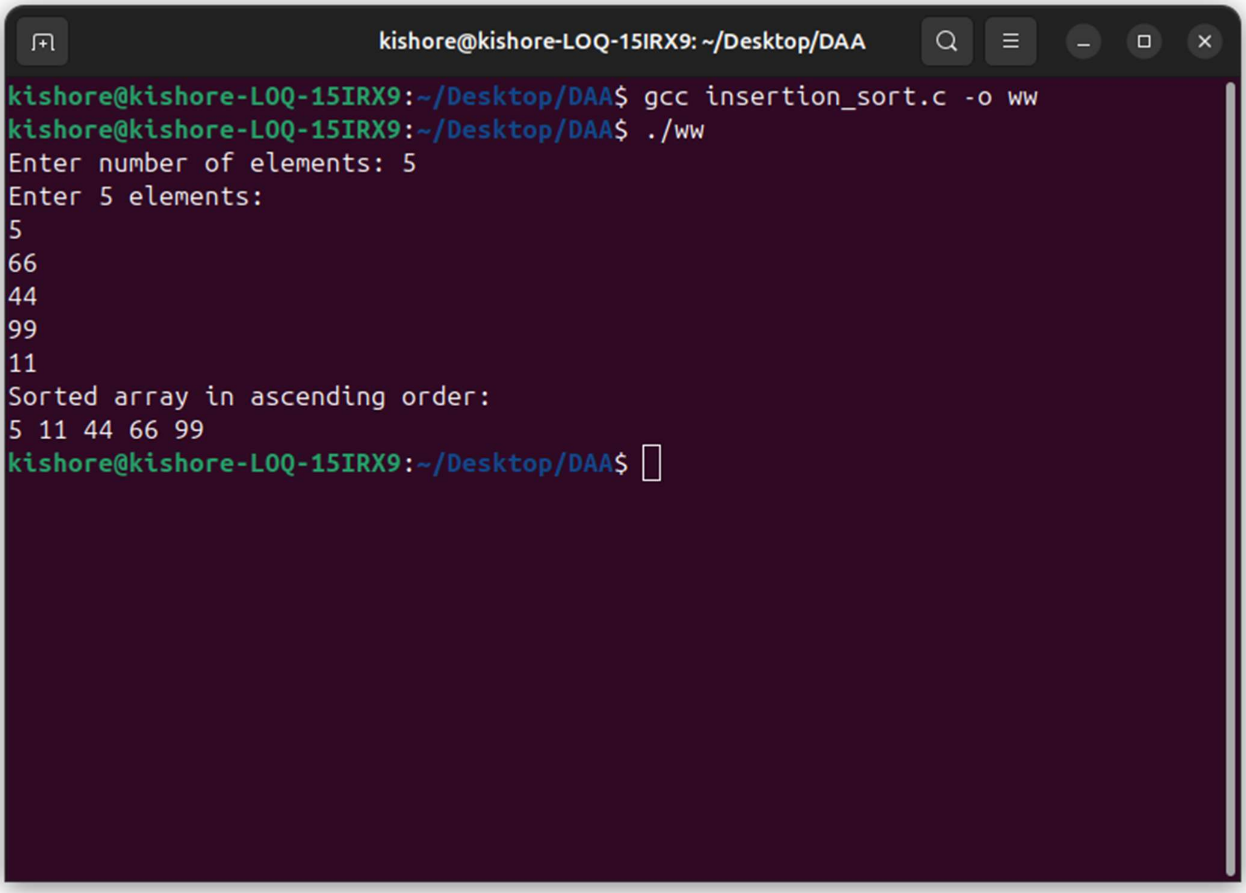
```
kishore@kishore-LOQ-15IRX9:~/Desktop/DAA$ gcc selection_sort.c -o qq
kishore@kishore-LOQ-15IRX9:~/Desktop/DAA$ ./qq
Enter number of elements: 5
Enter 5 elements:
77
12
34
65
77
Sorted array in ascending order:
12 34 65 77 77
kishore@kishore-LOQ-15IRX9:~/Desktop/DAA$ 
```

3.Write a c program for insertion sort

Code:

```
#include <stdio.h>

int main() {
    int n, i, j, key;
    int arr[100];

    printf("Enter number of elements: ");
    scanf("%d", &n);

    printf("Enter %d elements:\n", n);
    for (i = 0; i < n; i++) {
        scanf("%d", &arr[i]);
    }

    // Insertion Sort logic
    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;
    }

    printf("Sorted array in ascending order:\n");
    for (i = 0; i < n; i++) {
        printf("%d ", arr[i]);
    }
    printf("\n");

    return 0;
}
```

Output:



```
kishore@kishore-LOQ-15IRX9:~/Desktop/DAA$ gcc insertion_sort.c -o ww
kishore@kishore-LOQ-15IRX9:~/Desktop/DAA$ ./ww
Enter number of elements: 5
Enter 5 elements:
5
66
44
99
11
Sorted array in ascending order:
5 11 44 66 99
kishore@kishore-LOQ-15IRX9:~/Desktop/DAA$ 
```

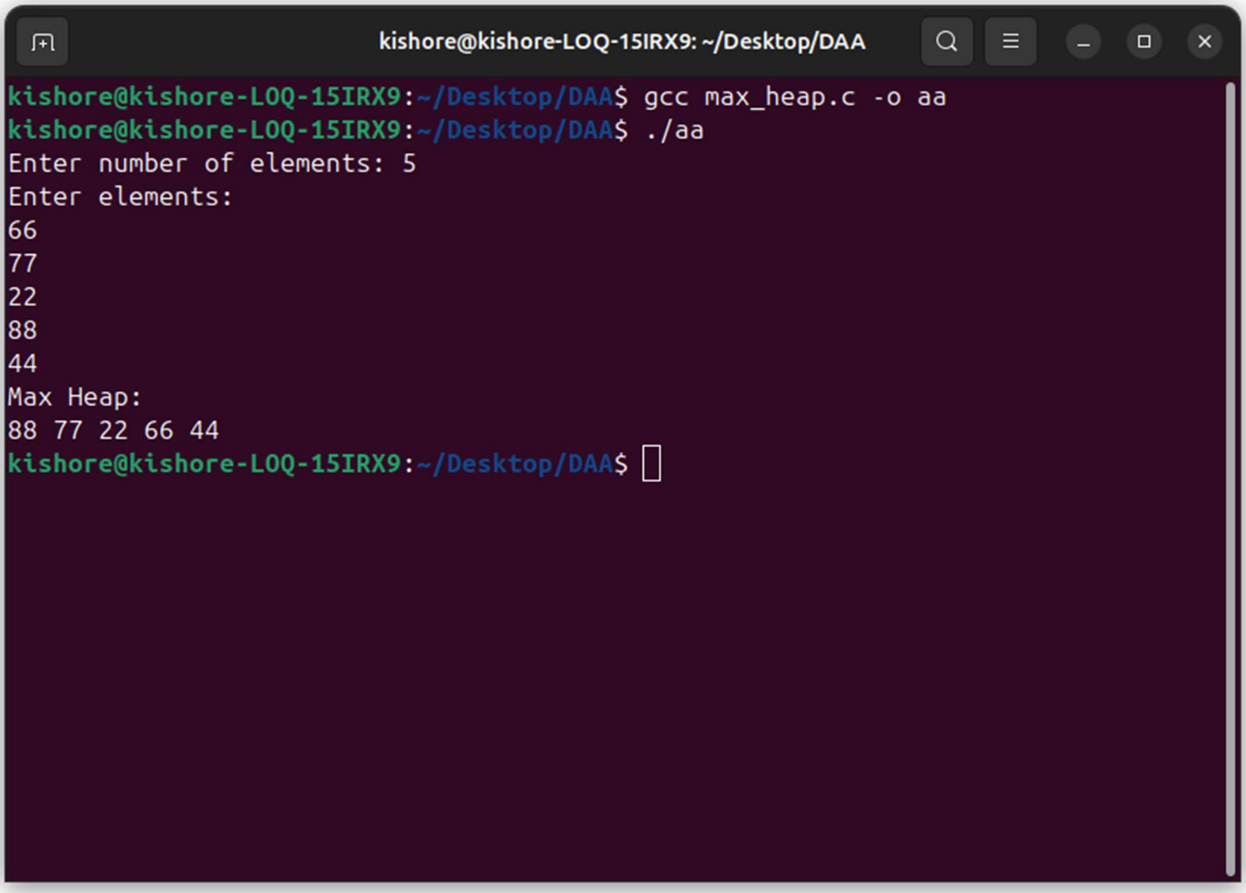
4. Write a c program for heap
a) max heap

Code:

```
#include <stdio.h>


int heap[100];

int size = 0;


void swap(int *a, int *b) {

    int temp = *a;

    *a = *b;

    *b = temp;

}
```

```
void insertMax(int value) {
    int i = size;
    heap[size++] = value;

    while (i != 0 && heap[(i - 1) / 2] < heap[i]) {
        swap(&heap[i], &heap[(i - 1) / 2]);
        i = (i - 1) / 2;
    }
}

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

int main() {
    int n, x;

    printf("Enter number of elements: ");
    scanf("%d", &n);

    printf("Enter elements:\n");
    for (int i = 0; i < n; i++) {
        scanf("%d", &x);
        insertMax(x);
    }

    printf("Max Heap:\n");
    display();
    return 0;
}
```

Output:



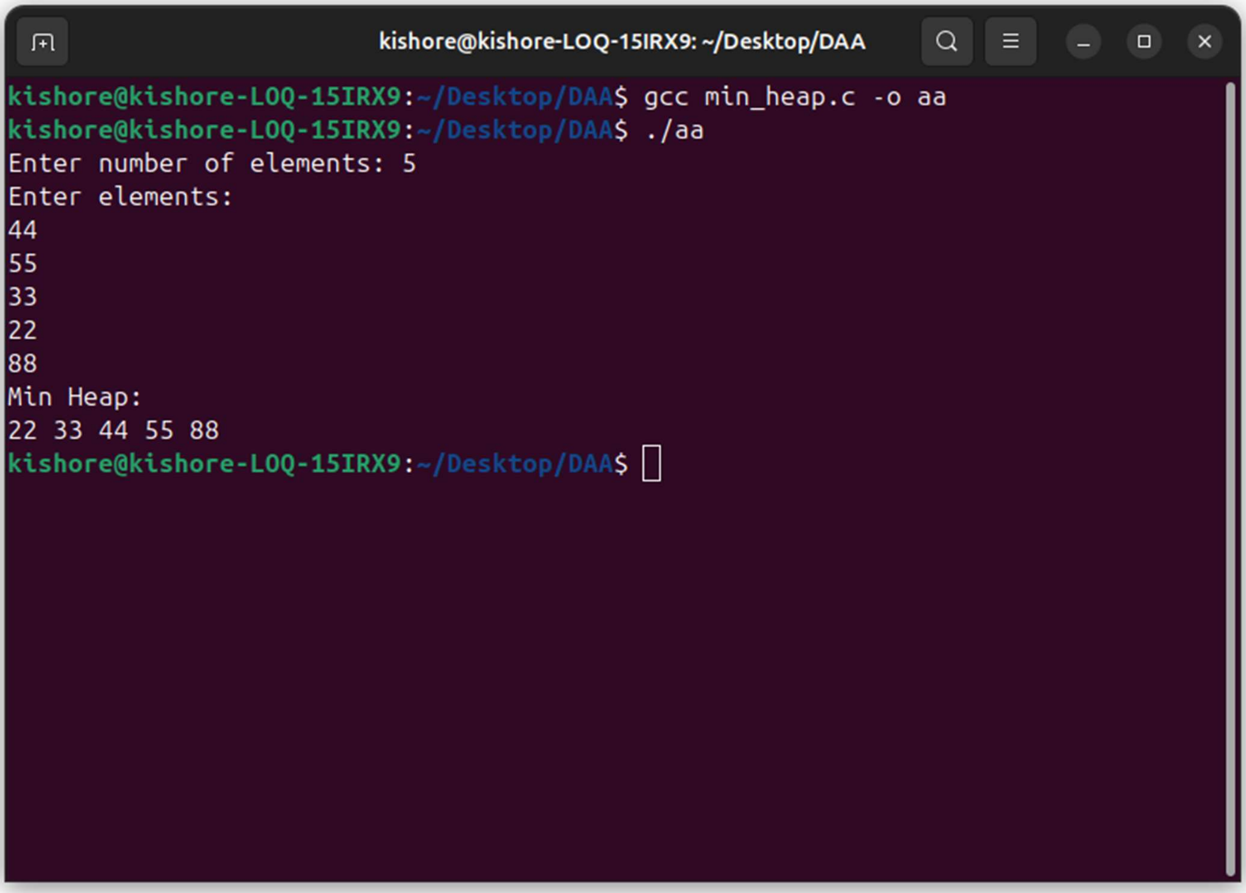
```
kishore@kishore-LOQ-15IRX9:~/Desktop/DAA$ gcc max_heap.c -o aa
kishore@kishore-LOQ-15IRX9:~/Desktop/DAA$ ./aa
Enter number of elements: 5
Enter elements:
66
77
22
88
44
Max Heap:
88 77 22 66 44
kishore@kishore-LOQ-15IRX9:~/Desktop/DAA$ 
```

b) mini heap

code:

```
#include <stdio.h>

int heap[100];
int size = 0;

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

void insertMin(int value) {
    int i = size;
    heap[size++] = value;

    while (i != 0 && heap[(i - 1) / 2] > heap[i]) {
        swap(&heap[i], &heap[(i - 1) / 2]);
        i = (i - 1) / 2;
    }
}

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

int main() {
    int n, x;

    printf("Enter number of elements: ");
    scanf("%d", &n);

    printf("Enter elements:\n");
    for (int i = 0; i < n; i++) {
        scanf("%d", &x);
        insertMin(x);
    }

    printf("Min Heap:\n");
    display();

    return 0;
}
```

Output:



```
kishore@kishore-LOQ-15IRX9:~/Desktop/DAA$ gcc min_heap.c -o aa
kishore@kishore-LOQ-15IRX9:~/Desktop/DAA$ ./aa
Enter number of elements: 5
Enter elements:
44
55
33
22
88
Min Heap:
22 33 44 55 88
kishore@kishore-LOQ-15IRX9:~/Desktop/DAA$ 
```

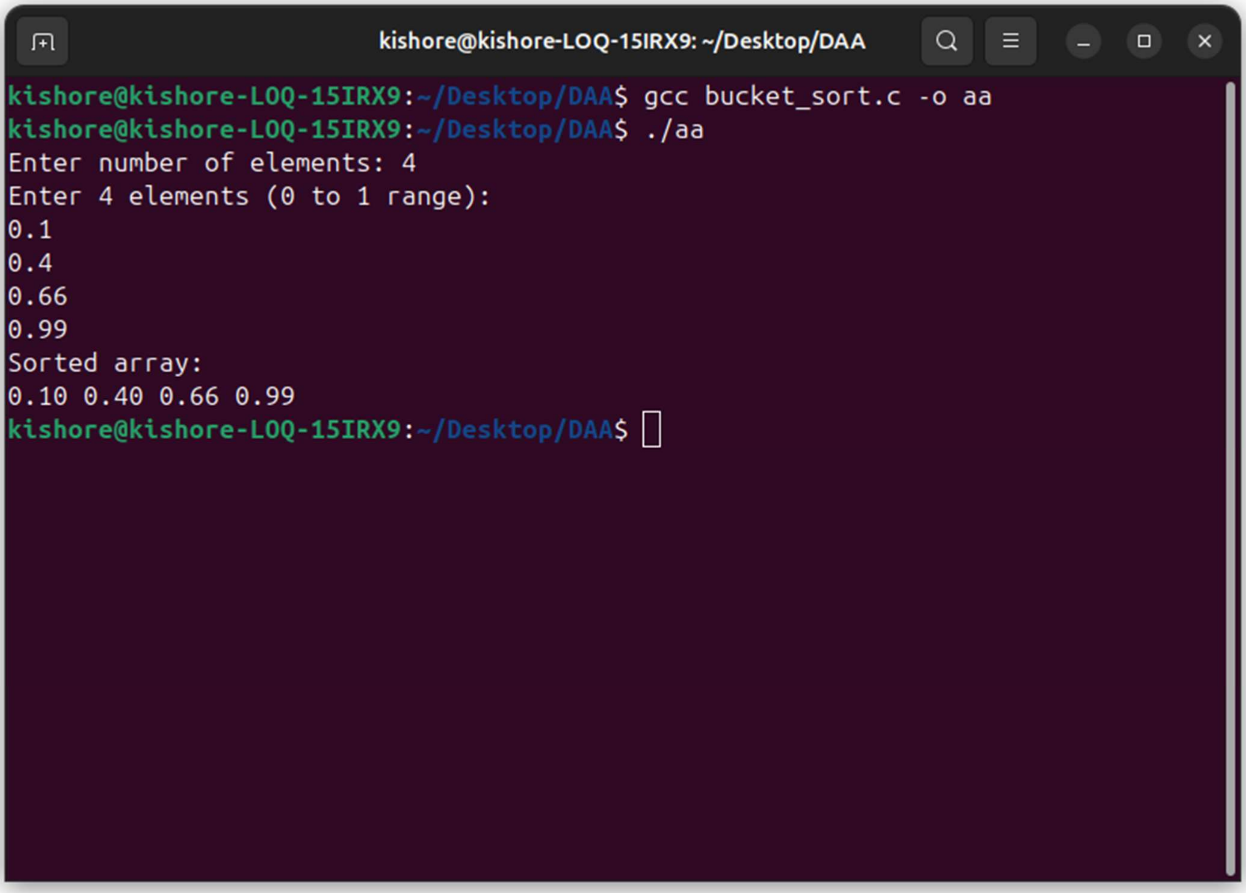
5.Write a c program bucket sort

Code:

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

void insertionSort(float bucket[], int n) {
    int i, j;
    float key;

    for (i = 1; i < n; i++) {
        key = bucket[i];
        j = i - 1;

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

```
int main() {
    int n, i, j;
    float arr[100];
    float bucket[10][100];
    int count[10] = {0};

    printf("Enter number of elements: ");
    scanf("%d", &n);

    printf("Enter %d elements (0 to 1 range):\n", n);
    for (i = 0; i < n; i++) {
        scanf("%f", &arr[i]);
    }

    for (i = 0; i < n; i++) {
        int index = n * arr[i];
        bucket[index][count[index]++] = arr[i];
    }

    for (i = 0; i < n; i++) {
        insertionSort(bucket[i], count[i]);
    }

    j = 0;
    for (i = 0; i < n; i++) {
        for (int k = 0; k < count[i]; k++) {
            arr[j++] = bucket[i][k];
        }
    }

    printf("Sorted array:\n");
    for (i = 0; i < n; i++) {
        printf("%.2f ", arr[i]);
    }
    printf("\n");

    return 0;
}
```

Output:



```
kishore@kishore-LOQ-15IRX9:~/Desktop/DAA$ gcc bucket_sort.c -o aa
kishore@kishore-LOQ-15IRX9:~/Desktop/DAA$ ./aa
Enter number of elements: 4
Enter 4 elements (0 to 1 range):
0.1
0.4
0.66
0.99
Sorted array:
0.10 0.40 0.66 0.99
kishore@kishore-LOQ-15IRX9:~/Desktop/DAA$
```