Name: Chepoori Sai Vivek

Class & Section: CSE – B

Roll No: CH.SC.U4CSE24108

1. Bubble Sort

CODE:

```c
#include <stdio.h>
// CH.SC.U4CSE24108 - Bubble Sort
void bubbleSort(int a[], int n) {
    int i, j, temp, swapped;
    for (i = 0; i < n - 1; i++) {
        swapped = 0;
        for (j = 0; j < n - i - 1; j++) {
            if (a[j] > a[j + 1]) {
                temp = a[j];
                a[j] = a[j + 1];
                a[j + 1] = temp;
                swapped = 1;
            }
        }
        if (!swapped)
            break;
    }
}
int main() {
    printf("CH.SC.U4CSE24108\n");
    int a[5] = {5, 1, 4, 2, 8};
    int n = 5, i;
    bubbleSort(a, n);
    for (i = 0; i < n; i++) {
        printf("%d ", a[i]);
    }
    return 0;
}
```

OUTPUT:

```
amma@amma06:~/Documents$ gcc p1.c -o p1
amma@amma06:~/Documents$ ./p1
CH.SC.U4CSE24108
1 2 4 5 8 amma@amma06:~/Documents$
amma@amma06:~/Documents$
```

2. Insertion Sort

CODE:

```c
#include <stdio.h>
// CH.SC.U4CSE24108 - Insertion Sort
void insertionSort(int a[], int n) {
    int i, key, j;
    for (i = 1; i < n; i++) {
        key = a[i];
        j = i - 1;

        while (j >= 0 && a[j] > key) {
            a[j + 1] = a[j];
            j--;
        }
        a[j + 1] = key;
    }
}
int main() {
    printf("CH.SC.U4CSE24108\n");
    int a[5] = {12, 11, 13, 5, 6};
    int n = 5, i;
    insertionSort(a, n);
    for (i = 0; i < n; i++)
        printf("%d ", a[i]);
    return 0;
}
```

OUTPUT:

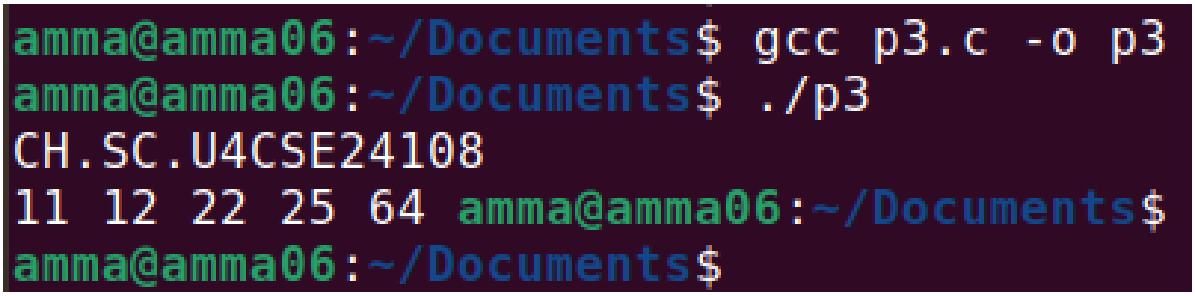
```
amma@amma06:~/Documents$ gcc p2.c -o p2
amma@amma06:~/Documents$ ./p2
CH.SC.U4CSE24108
5 6 11 12 13 amma@amma06:~/Documents$
amma@amma06:~/Documents$
```

3. Selection Sort

CODE:

```c
#include <stdio.h>
// CH.SC.U4CSE24108 - Selection Sort
void selectionSort(int a[], int n) {
    int i, j, min, temp;
    for (i = 0; i < n - 1; i++) {
        min = i;
        for (j = i + 1; j < n; j++) {
            if (a[j] < a[min])
                min = j;
        }
        temp = a[i];
        a[i] = a[min];
        a[min] = temp;
    }
}
int main() {
    printf("CH.SC.U4CSE24108\n");
    int a[5] = {64, 25, 12, 22, 11};
    int n = 5, i;
    selectionSort(a, n);
    for (i = 0; i < n; i++)
        printf("%d ", a[i]);
    return 0;
}
```

OUTPUT:

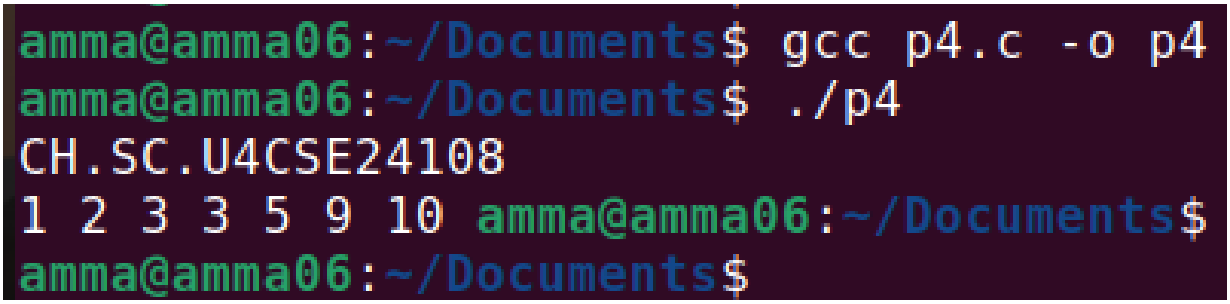

amma@amma06:~/Documents$ gcc p3.c -o p3
amma@amma06:~/Documents$ ./p3
CH.SC.U4CSE24108
11 12 22 25 64 amma@amma06:~/Documents$
amma@amma06:~/Documents$

4. Bucket Sort

CODE:

```c
#include <stdio.h>
// CH.SC.U4CSE24108 - Bucket Sort
void bucketSort(int a[], int n) {
    int bucket[100] = {0};
    int i;
    for (i = 0; i < n; i++)
        bucket[a[i]]++;
    int index = 0;
    for (i = 0; i < 100; i++) {
        while (bucket[i] > 0) {
            a[index++] = i;
            bucket[i]--;
        }
    }
}
int main() {
    printf("CH.SC.U4CSE24108\n");
    int a[7] = {10, 3, 5, 2, 9, 3, 1};
    int n = 7, i;
    bucketSort(a, n);
    for (i = 0; i < n; i++)
        printf("%d ", a[i]);
    return 0;
}
```

OUTPUT:

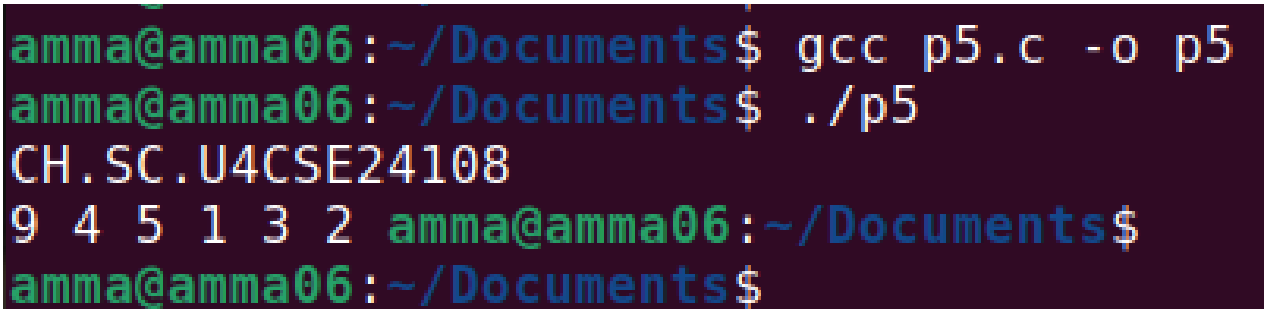

amma@amma06:~/Documents$ gcc p4.c -o p4
amma@amma06:~/Documents$ ./p4
CH.SC.U4CSE24108
1 2 3 3 5 9 10 amma@amma06:~/Documents$
amma@amma06:~/Documents$

5. MAX HEAP

CODE:

```c
#include <stdio.h>
// CH.SC.U4CSE24108 - Max Heap
void maxHeapify(int a[], int n, int i) {
    int largest = i;
    int left = 2*i + 1;
    int right = 2*i + 2;
    int temp;
    if (left < n && a[left] > a[largest])
        largest = left;
    if (right < n && a[right] > a[largest])
        largest = right;
    if (largest != i) {
        temp = a[i];
        a[i] = a[largest];
        a[largest] = temp;
        maxHeapify(a, n, largest);
    }
}
void buildMaxHeap(int a[], int n) {
    int i;
    for (i = n/2 - 1; i >= 0; i--)
        maxHeapify(a, n, i);
}
int main() {
    printf("CH.SC.U4CSE24108\n");
    int a[6] = {3, 9, 2, 1, 4, 5};
    int n = 6, i;
    buildMaxHeap(a, n);
    for (i = 0; i < n; i++)
        printf("%d ", a[i]);
    return 0;
}
```

OUTPUT:

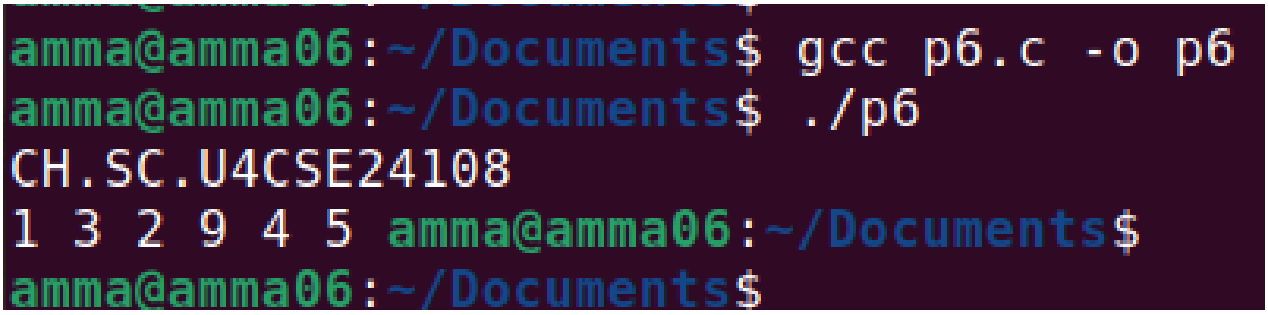

amma@amma06:~/Documents$ gcc p5.c -o p5
amma@amma06:~/Documents$ ./p5
CH.SC.U4CSE24108
9 4 5 1 3 2 amma@amma06:~/Documents$
amma@amma06:~/Documents$

6. MIN HEAP

CODE:

```c
#include <stdio.h>
// CH.SC.U4CSE24108 - Min Heap
void minHeapify(int a[], int n, int i) {
    int smallest = i;
    int left = 2*i + 1;
    int right = 2*i + 2;
    int temp;
    if (left < n && a[left] < a[smallest])
        smallest = left;
    if (right < n && a[right] < a[smallest])
        smallest = right;
    if (smallest != i) {
        temp = a[i];
        a[i] = a[smallest];
        a[smallest] = temp;
        minHeapify(a, n, smallest);
    }
}
void buildMinHeap(int a[], int n) {
    int i;
    for (i = n/2 - 1; i >= 0; i--)
        minHeapify(a, n, i);
}
int main() {
    printf("CH.SC.U4CSE24108\n");
    int a[6] = {3, 9, 2, 1, 4, 5};
    int n = 6, i;
    buildMinHeap(a, n);
    for (i = 0; i < n; i++)
        printf("%d ", a[i]);
    return 0;
}
```

OUTPUT:

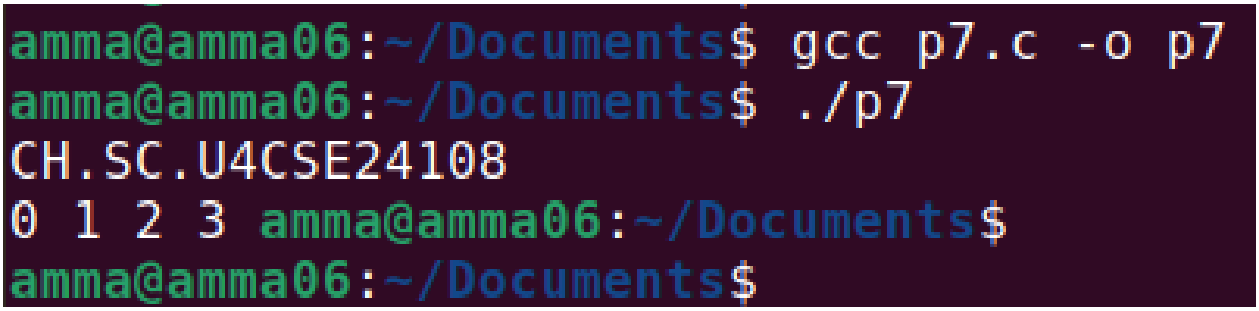

amma@amma06:~/Documents$ gcc p6.c -o p6
amma@amma06:~/Documents$ ./p6
CH.SC.U4CSE24108
1 3 2 9 4 5 amma@amma06:~/Documents$
amma@amma06:~/Documents$

7. BFS

CODE:

```c
#include <stdio.h>
#define MAX 100
// CH.SC.U4CSE24108 - BFS
int queue[MAX], front = 0, rear = 0;
void enqueue(int x) { queue[rear++] = x; }
int dequeue() { return queue[front++]; }
void bfs(int graph[MAX][MAX], int n, int start) {
    int visited[MAX];
    int i, node;
    for (i = 0; i < n; i++)
        visited[i] = 0;
    enqueue(start);
    visited[start] = 1;
    while (front != rear) {
        node = dequeue();
        printf("%d ", node);
        for (i = 0; i < n; i++) {
            if (graph[node][i] == 1 && !visited[i]) {
                visited[i] = 1;
                enqueue(i);
            }
        }
    }
}
int main() {
    printf("CH.SC.U4CSE24108\n");
    int n = 4;
    int graph[MAX][MAX] = {
        {0,1,1,0},
        {1,0,1,1},
        {1,1,0,0},
        {0,1,0,0}
    };
    bfs(graph, n, 0);
    return 0;
}
```

OUTPUT:


amma@amma06:~/Documents$ gcc p7.c -o p7
amma@amma06:~/Documents$ ./p7
CH.SC.U4CSE24108
0 1 2 3 amma@amma06:~/Documents$
amma@amma06:~/Documents$

8. DFS

CODE:

```c
#include <stdio.h>
#define MAX 100
// CH.SC.U4CSE24108 - DFS
void dfs(int graph[MAX][MAX], int visited[], int n, int node) {
    int i;
    visited[node] = 1;
    printf("%d ", node);
    for (i = 0; i < n; i++) {
        if (graph[node][i] == 1 && !visited[i])
            dfs(graph, visited, n, i);
    }
}
int main() {
    printf("CH.SC.U4CSE24108\n");
    int n = 4, i;
    int visited[MAX];
    int graph[MAX][MAX] = {
        {0,1,1,0},
        {1,0,1,1},
        {1,1,0,0},
        {0,1,0,0}
    };
    for (i = 0; i < n; i++)
        visited[i] = 0;
    dfs(graph, visited, n, 0);
    return 0;
}
```

OUTPUT:

```
amma@amma06:~/Documents$ gcc p8.c -o p8
amma@amma06:~/Documents$ ./p8
CH.SC.U4CSE24108
0 1 2 3 amma@amma06:~/Documents$
amma@amma06:~/Documents$
```

****