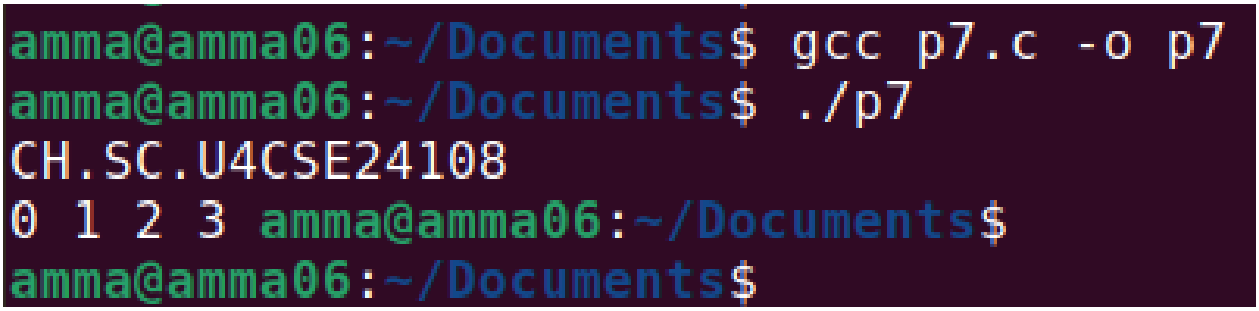
7. BFS

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$
```

****