

S.No: 1	Exp. Name: <i>Project Module</i>	Date: 2024-06-13
---------	----------------------------------	------------------

Aim:

Project Module

Source Code:

```
hello.c
```

```

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

#define MAX_NODES 100

// Adjacency Matrix Representation
int graph[MAX_NODES][MAX_NODES];
int visited[MAX_NODES];
int nodeCount;

// Data Collection Module
void collectData() {
    printf("Enter number of nodes: ");
    scanf("%d", &nodeCount);

    printf("Enter adjacency matrix:\n");
    for (int i = 0; i < nodeCount; i++) {
        for (int j = 0; j < nodeCount; j++) {
            scanf("%d", &graph[i][j]);
        }
    }
}

// Graph Construction Module
void initializeGraph() {
    for (int i = 0; i < MAX_NODES; i++) {
        for (int j = 0; j < MAX_NODES; j++) {
            graph[i][j] = 0;
        }
        visited[i] = 0;
    }
}

// Community Detection Module
void DFS(int node) {
    printf("%d ", node);
    visited[node] = 1;

    for (int i = 0; i < nodeCount; i++) {
        if (graph[node][i] == 1 && !visited[i]) {
            DFS(i);
        }
    }
}

void detectCommunities() {
    printf("Communities:\n");
    for (int i = 0; i < nodeCount; i++) {
        if (!visited[i]) {
            DFS(i);
            printf("\n");
        }
    }
}

```

```

    printf("Degree Centrality:\n");
    for (int i = 0; i < nodeCount; i++) {
        int degree = 0;
        for (int j = 0; j < nodeCount; j++) {
            if (graph[i][j] == 1) {
                degree++;
            }
        }
        printf("Node %d: %d\n", i, degree);
    }
}

// Connectivity Analysis Module
void connectivityAnalysis() {
    printf("Connectivity Analysis:\n");
    for (int i = 0; i < nodeCount; i++) {
        for (int j = 0; j < nodeCount; j++) {
            if (graph[i][j] == 1) {
                printf("Node %d is connected to Node %d\n", i, j);
            }
        }
    }
}

// Visualization and Reporting Module
void visualizeGraph() {
    printf("Adjacency Matrix:\n");
    for (int i = 0; i < nodeCount; i++) {
        for (int j = 0; j < nodeCount; j++) {
            printf("%d ", graph[i][j]);
        }
        printf("\n");
    }
}

int main() {
    initializeGraph();
    collectData();
    visualizeGraph();
    detectCommunities();
    measureCentrality();
    connectivityAnalysis();

    return 0;
}

```

Execution Results - All test cases have succeeded!

Test Case - 1
User Output
Hello World