Experiment 22: Graph Traversal using DFS

Aim:
To write a C program to perform Depth First Search (DFS) traversal of a graph.

Algorithm:

1. Start the program.
2. Represent the graph using adjacency matrix/list.
3. Use recursion or a stack for DFS.
4. Mark the starting node as visited and print it.
5. Recursively visit all unvisited neighbors.
6. Stop.

Code:

```
#include <stdio.h>
#define SIZE 10

void dfs(int adj[SIZE][SIZE], int visited[SIZE], int n, int v) {
   printf("%d ", v);
   visited[v] = 1;
   for (int i = 0; i < n; i++) {
      if (adj[v][i] && !visited[i]) {
         dfs(adj, visited, n, i);
      }
   }
}

int main() {
   int n = 4;
   int adj[SIZE][SIZE] = {
      {0,1,1,0},
      {1,0,0,1},
      {1,0,0,1},
      {0,1,1,0}
```

```
    };
    int visited[SIZE] = {0};
    printf("DFS starting from vertex 0: ");
    dfs(adj, visited, n, 0);
    return 0;
}
```

Sample Output:

```
DFS starting from vertex 0: 0 1 3 2

=== Code Execution Successful ===
```

Result:
The program successfully traverses a graph using DFS.