DFS CODE:

```
#include <stdio.h>

#define MAX 10

int visited[MAX];
int adj[MAX][MAX];
int n;

// DFS function
void DFS(int v) {
   visited[v] = 1;
   printf("%d ", v);

   for (int i = 0; i < n; i++) {
      if (adj[v][i] == 1 && !visited[i]) {
         DFS(i);
      }
   }
}

int main() {
   printf("Enter number of vertices: ");
   scanf("%d", &n);

   printf("Enter adjacency matrix:\n");
   for (int i = 0; i < n; i++) {
```

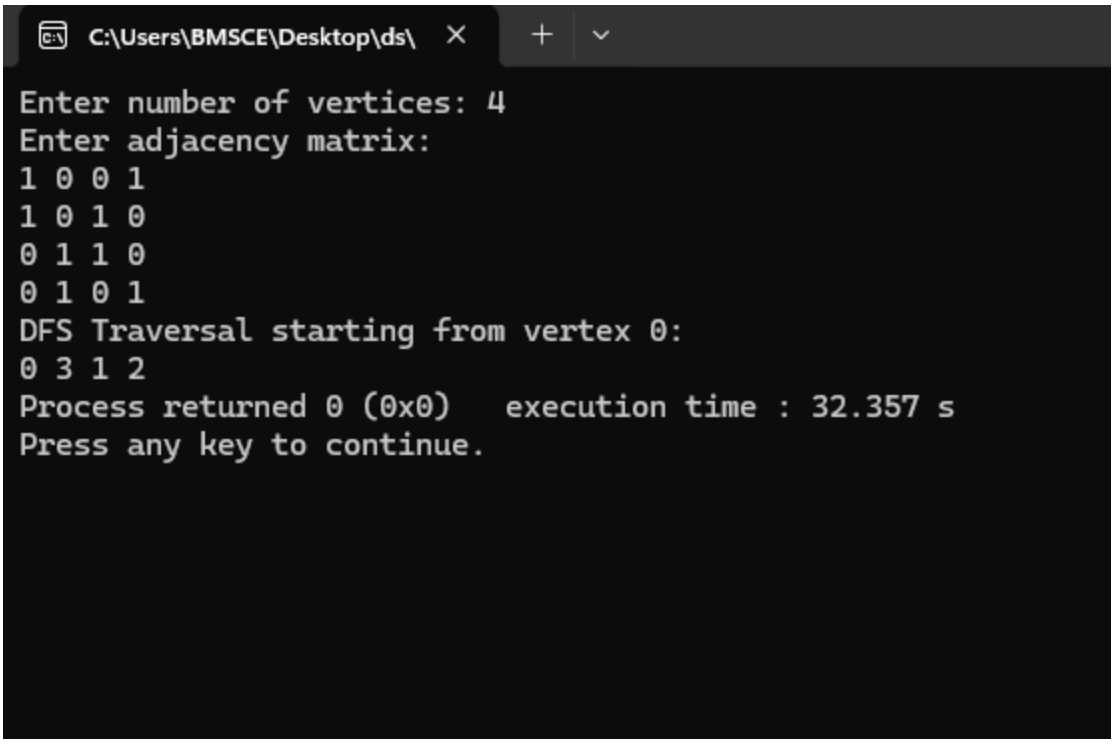
```
        for (int j = 0; j < n; j++) {
            scanf("%d", &adj[i][j]);
        }
    }

    // Initialize visited array
    for (int i = 0; i < n; i++)
        visited[i] = 0;

    printf("DFS Traversal starting from vertex 0:\n");
    DFS(0);

    return 0;
}
```

OUTPUT:



OBSERVATION:

Page No:
Date.

22/12/25

Prg-9b

WAP to traverse a graph using DFS method.

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

int visited[MAX];
int adj[MAX][MAX];
int n;

void DFS(int v) {
    visited[v] = 1;
    printf("%d", v);

    for (int i=0; i<n; i++) {
        if (adj[v][i] == 1 && !visited[i]) {
            DFS(i);
        }
    }
}

int main() {
    printf("Enter number of vertices: ");
    scanf("%d", &n);

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

```
for (int i = 0; i < n; i++)
    visited[i] = 0;

    printf("DFS Traversal starting from
    vertex 0: \n");
    DFS(0);

    return 0;
}
```

Output :-

```
Enter number of vertices : 4
Enter adjacency matrix :
1 0 0 1
1 0 1 0
0 1 1 0
0 1 0 1

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

22/12/25