

```
#include <stdio.h>
```

```
#include <stdlib.h>
```

```
int Adjmat[4][4] = {
```

```
    {0, 0, 1, 0}, // Node 1 connected to Node 3
```

```
    {1, 0, 0, 0}, // Node 2 connected to Node 1
```

```
    {0, 0, 0, 1}, // Node 3 connected to Node 4
```

```
    {0, 1, 0, 0} // Node 4 connected to Node 2
```

```
};
```

```
int visit[4] = {0, 0, 0, 0}; // Array to keep track of visited nodes
```

```
int stack[4];
```

```
int top = -1;
```

```
int isEmpty() {
```

```
    return top == -1;
```

```
}
```

```

int isFull() {
    return top == 4 - 1;
}

void push(int element) {
    if (isFull()) {
        printf("Stack is full\n");
        return;
    }
    stack[++top] = element;
}

int pop() {
    if (isEmpty()) {
        printf("Stack is empty\n");
        return -1;
    }
    return stack[top--];
}

void dfs(int startNode) {
    push(startNode);
    printf("DFS Traversal: ");
    while (!isEmpty()) {
        int currentNode = pop();
        if (!visit[currentNode]) {
            printf("%d ", currentNode + 1); // Printing nodes as 1-based index
            visit[currentNode] = 1;
        }
    }
}

```

```
// Push all adjacent vertices of the current node into the stack
for (int i = 0; i < 4; i++) {
    if (Adjmat[currentNode][i] == 1 && !visit[i]) {
        push(i);
    }
}
}
}
printf("\n");
}

int main() {
    dfs(0); // Start DFS from node 1 (0-based index)
    return 0;
}
```