

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

```
#define MAX_VERTICES 10
```

```
// Function to input the graph and store it as an adjacency matrix
```

```
void inputGraph(int graph[MAX_VERTICES][MAX_VERTICES], int vertices, int  
edges) {
```

```
    int i, j, v1, v2;
```

```
    // Initialize the adjacency matrix
```

```
    for (i = 0; i < vertices; i++) {
```

```
        for (j = 0; j < vertices; j++) {
```

```
            graph[i][j] = 0;
```

```
        }
```

```
    }
```

```
    // Input edges and update the adjacency matrix
```

```
    printf("Enter %d edges (vertex1 vertex2):\n", edges);
```

```
    for (i = 0; i < edges; i++) {
```

```
        scanf("%d %d", &v1, &v2);
```

```
        graph[v1][v2] = 1;
```

```
        graph[v2][v1] = 1; // Assuming an undirected graph
```

```
    }
```

```
}
```

```
// Function to display the adjacency matrix
```

```
void displayGraph(int graph[MAX_VERTICES][MAX_VERTICES], int vertices) {
```

```
    int i, j;
```

```
    printf("\nAdjacency Matrix:\n");
```

```
    // Display the column indices
```

```
    printf(" ");
```

```
    for (i = 0; i < vertices; i++) {
```

```
        printf("%d ", i);
```

```
    }
```

```
    printf("\n");
```

```
    // Display the matrix
```

```
    for (i = 0; i < vertices; i++) {
```

```

        printf("%d ", i);
        for (j = 0; j < vertices; j++) {
            printf("%d ", graph[i][j]);
        }
        printf("\n");
    }
}

int main() {
    int vertices, edges;

    // Input the number of vertices and edges
    printf("Enter the number of vertices: ");
    scanf("%d", &vertices);
    printf("Enter the number of edges: ");
    scanf("%d", &edges);

    if (vertices > MAX_VERTICES) {
        printf("Number of vertices exceeds the maximum limit.\n");
        return 1;
    }

    int graph[MAX_VERTICES][MAX_VERTICES];

    // Input the graph and store it as an adjacency matrix
    inputGraph(graph, vertices, edges);

    // Display the adjacency matrix
    displayGraph(graph, vertices);

    return 0;
}

```

Enter the number of vertices: 5

Enter the number of edges: 6

Enter 6 edges (vertex1 vertex2):

0 1

0 2

1 3
2 4
3 4
1 2

Adjacency Matrix:

| | 0 | 1 | 2 | 3 | 4 |
|---|---|---|---|---|---|
| 0 | 0 | 0 | 1 | 1 | 0 |
| 1 | 1 | 1 | 0 | 1 | 1 |
| 2 | 1 | 1 | 0 | 0 | 1 |
| 3 | 0 | 1 | 0 | 0 | 1 |
| 4 | 0 | 0 | 1 | 1 | 0 |