9) Implement functiontoprintIn-Degree,Out-Degreeandtodisplaythatadjacency matric

```
#include <stdio.h>

#define MAX_VERTICES 10

void printAdjacencyMatrix(int matrix[][MAX_VERTICES], int n) {
    printf("Adjacency Matrix:\n");
    for (int i = 0; i < n; i++) {
        for (int j = 0; j < n; j++) {
            printf("%d ", matrix[i][j]);
        }
        printf("\n");
    }
}

void printInDegree(int matrix[][MAX_VERTICES], int n) {
    printf("In-Degree of each vertex:\n");
    for (int i = 0; i < n; i++) {
        int inDegree = 0;
        for (int j = 0; j < n; j++) {
            inDegree += matrix[j][i];
        }
        printf("Vertex %d: %d\n", i, inDegree);
    }
}

void printOutDegree(int matrix[][MAX_VERTICES], int n) {
    printf("Out-Degree of each vertex:\n");
    for (int i = 0; i < n; i++) {
        int outDegree = 0;
        for (int j = 0; j < n; j++) {
            outDegree += matrix[i][j];
        }
        printf("Vertex %d: %d\n", i, outDegree);
    }
}

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

    int adjMatrix[MAX_VERTICES][MAX_VERTICES];

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

    printAdjacencyMatrix(adjMatrix, n);
    printInDegree(adjMatrix, n);
    printOutDegree(adjMatrix, n);

    return 0;
}
```

Output:

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

Adjacency Matrix:
0 1 1 0
1 0 1 1
1 1 0 1
0 1 1 0

In-Degree of each vertex:
Vertex 0: 2
Vertex 1: 3
Vertex 2: 3
Vertex 3: 2

Out-Degree of each vertex:
```

```
Vertex 0: 2
Vertex 1: 3
Vertex 2: 3
Vertex 3: 2
```