SC1007 Data Structures and Algorithms **2022/23 Semester 2**

Lab 6: Graph Representation

School of Computer Science and Engineering *Nanyang Technological University*

Q1 Write a function adjM2adjL() to convert an adjacency matrix to an adjacency list. The structure of a graph is given below.

```c
enum GraphType {ADJ_MATRIX, ADJ_LIST}; // Types of Graph Representation

typedef struct _listnode
{
    int vertex;
  struct _listnode *next;
} ListNode;

union GraphForm{
    int **matrix;
    ListNode **list;
};

typedef struct _graph{
    int V;
    int E;
    enum GraphType type;
    union GraphForm adj;
}Graph;
```

the vertices are named from 1 to $|V|$.

The function prototype is given as follows:

```c
void adjM2adjL(Graph *g);
```

Q2 Write a function adjL2adjM() to convert an adjacency list to an adjacency matrix. Please reuse the work down in Q1. The function prototype is given as follows:

```c
void adjL2adjM(Graph *g);
```

Q3 The degree of a vertex v of a graph is the number of edges incident on v. Write a function calDegreeV() to compute vertex degrees using adjacent lists and using adjacency matrix. Please reuse the work done in Q1 and Q2.

```c
void calDegreeV(Graph g, int *degreeV)
```

Q4 Write a function BFS() to do a breadth first search from a input vertex v and print out the visited vertices in the order of visiting. The labels of v are from 1 to $|V|$. The algorithm will visit the neighbor nodes in ascending order. The function prototype is given as follows:

```c
void BFS(Graph g, int v)
```

Remark Please make sure that your program will not be crashed by continuously converting between two representation forms and the degree of vertices is correctly computed in every conversion.