**6-1 邻接矩阵存储图的深度优先遍历 (20分)**

试实现邻接矩阵存储图的深度优先遍历。

函数接口定义：

**void** **DFS**( MGraph Graph, Vertex V, **void** (\*Visit)(Vertex) );

其中MGraph是邻接矩阵存储的图，定义如下：

**typedef** **struct** **GNode** \***PtrToGNode**;

**struct** **GNode**{

**int** Nv; */\* 顶点数 \*/*

**int** Ne; */\* 边数 \*/*

WeightType G[MaxVertexNum][MaxVertexNum]; */\* 邻接矩阵 \*/*

};

**typedef** PtrToGNode MGraph; */\* 以邻接矩阵存储的图类型 \*/*

函数DFS应从第V个顶点出发递归地深度优先遍历图Graph，遍历时用裁判定义的函数Visit访问每个顶点。当访问邻接点时，要求按序号递增的顺序。题目保证V是图中的合法顶点。

裁判测试程序样例：

**#include <stdio.h>**

**typedef** **enum** {false, true} **bool**;

**#define MaxVertexNum 10 */\* 最大顶点数设为10 \*/***

**#define INFINITY 65535 */\* ∞设为双字节无符号整数的最大值65535\*/***

**typedef** **int** Vertex; */\* 用顶点下标表示顶点,为整型 \*/*

**typedef** **int** WeightType; */\* 边的权值设为整型 \*/*

**typedef** **struct** **GNode** \***PtrToGNode**;

**struct** **GNode**{

**int** Nv; */\* 顶点数 \*/*

**int** Ne; */\* 边数 \*/*

WeightType G[MaxVertexNum][MaxVertexNum]; */\* 邻接矩阵 \*/*

};

**typedef** PtrToGNode MGraph; */\* 以邻接矩阵存储的图类型 \*/*

**bool** Visited[MaxVertexNum]; */\* 顶点的访问标记 \*/*

MGraph **CreateGraph**(); */\* 创建图并且将Visited初始化为false；裁判实现，细节不表 \*/*

**void** **Visit**( Vertex V )

{

printf(" %d", V);

}

**void** **DFS**( MGraph Graph, Vertex V, **void** (\*Visit)(Vertex) );

**int** **main**()

{

MGraph G;

Vertex V;

G = CreateGraph();

scanf("%d", &V);

printf("DFS from %d:", V);

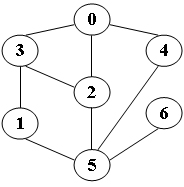
DFS(G, V, Visit);

**return** 0;

}

*/\* 你的代码将被嵌在这里 \*/*

输入样例：给定图如下



5

输出样例：

DFS from 5: 5 1 3 0 2 4 6