[문제 1]

```
typedef struct Edge
{
  int endpoint; // weight
  char vertex1, vertex2;
}Edge;

typedef struct IncidentEdge
{
  int edge_num;
  IncidentEdge* next;
}IncidentEdge;

typedef struct Vertex
{
  char name;
  IncidentEdge* incidentedge;
}Vertex;

typedef struct Graph
{
  Vertex vertex[8];
  Edge edge[22];
}Graph;
```

```
void initi(Graph* graph)
{
int i;


for(i=0;i<MAX;i++)
{
graph->vertex[i].name = i + '1';

graph->vertex[i].incidentedge = (IncidentEdge*)malloc(sizeof(IncidentEdge));
memset(graph->vertex[i].incidentedge, 0, sizeof(IncidentEdge));
graph->vertex[i].incidentedge->edge_num = -1; // header check
}

memset(graph->edge, -1, sizeof(graph->edge));

graph->edge[0].endpoint = 1;
graph->edge[0].vertex1 = graph->vertex[0].name;
graph->edge[0].vertex2 = graph->vertex[1].name;


insert_vertexlist(graph, 0,  0);
insert_vertexlist(graph, 0,  1);

graph->edge[1].endpoint = 1;
graph->edge[1].vertex1 = graph->vertex[0].name;
graph->edge[1].vertex2 = graph->vertex[2].name;

insert_vertexlist(graph, 1,  0);
insert_vertexlist(graph, 1,  2);

graph->edge[2].endpoint = 1;
graph->edge[2].vertex1 = graph->vertex[0].name;
graph->edge[2].vertex2 = graph->vertex[3].name;

insert_vertexlist(graph, 2,  0);
insert_vertexlist(graph, 2,  3);

graph->edge[3].endpoint = 2;
graph->edge[3].vertex1 = graph->vertex[0].name;
graph->edge[3].vertex2 = graph->vertex[5].name;
```

```
insert_vertexlist(graph, 3,  0);
insert_vertexlist(graph, 3,  5);

graph->edge[4].endpoint = 1;
graph->edge[4].vertex1 = graph->vertex[1].name;
graph->edge[4].vertex2 = graph->vertex[2].name;

insert_vertexlist(graph, 4,  1);
insert_vertexlist(graph, 4,  2);

graph->edge[5].endpoint = 4;
graph->edge[5].vertex1 = graph->vertex[2].name;
graph->edge[5].vertex2 = graph->vertex[4].name;

insert_vertexlist(graph, 5,  2);
insert_vertexlist(graph, 5,  4);

graph->edge[6].endpoint = 4;
graph->edge[6].vertex1 = graph->vertex[4].name;
graph->edge[6].vertex2 = graph->vertex[4].name;

insert_vertexlist(graph, 6,  4);

graph->edge[7].endpoint = 3;
graph->edge[7].vertex1 = graph->vertex[4].name;
graph->edge[7].vertex2 = graph->vertex[5].name;

insert_vertexlist(graph, 7,  4);
insert_vertexlist(graph, 7,  5);

}
```

[문제 2]

```
typedef struct Edge
{
  int endpoint; // weight
  char vertex1, vertex2;
}Edge;

typedef struct Vertex
{
  char name;
}Vertex;

typedef struct Graph
{
  Vertex vertex[8];
  Edge edge[22];
  int adjacencyMatrix[8][8];
}Graph;



void initi(Graph* graph)
{
int i;

for(i=0;i<MAX;i++)
{
  graph->vertex[i].name = i + '1';
}

memset(graph->edge, -1, sizeof(graph->edge));
memset(graph->adjacencyMatrix, -1, sizeof(graph->adjacencyMatrix));

graph->edge[0].endpoint = 1;
graph->edge[0].vertex1 = '1';
graph->edge[0].vertex2 = '2';
graph->adjacencyMatrix[0][1] = 0;
graph->adjacencyMatrix[1][0] = 0;

graph->edge[1].endpoint = 1;
graph->edge[1].vertex1 = '1';
graph->edge[1].vertex2 = '3';
```

```
graph->adjacencyMatrix[0][2] = 1;
graph->adjacencyMatrix[2][0] = 1;

graph->edge[2].endpoint = 1;
graph->edge[2].vertex1 = '1';
graph->edge[2].vertex2 = '4';
graph->adjacencyMatrix[0][3] = 2;
graph->adjacencyMatrix[3][0] = 2;

graph->edge[3].endpoint = 2;
graph->edge[3].vertex1 = '1';
graph->edge[3].vertex2 = '6';
graph->adjacencyMatrix[0][5] = 3;
graph->adjacencyMatrix[5][0] = 3;

graph->edge[4].endpoint = 1;
graph->edge[4].vertex1 = '2';
graph->edge[4].vertex2 = '3';
graph->adjacencyMatrix[1][2] = 4;
graph->adjacencyMatrix[2][1] = 4;

graph->edge[5].endpoint = 4;
graph->edge[5].vertex1 = '3';
graph->edge[5].vertex2 = '5';
graph->adjacencyMatrix[2][4] = 5;
graph->adjacencyMatrix[4][2] = 5;

graph->edge[6].endpoint = 4;
graph->edge[6].vertex1 = '5';
graph->edge[6].vertex2 = '5';
graph->adjacencyMatrix[4][4] = 6;

graph->edge[7].endpoint = 3;
graph->edge[7].vertex1 = '5';
graph->edge[7].vertex2 = '6';
graph->adjacencyMatrix[4][5] = 7;
graph->adjacencyMatrix[5][4] = 7;

}
```