```cpp
#include<iostream>
using namespace std;

//-----------------------------------------------------有向图的十字链表法
#define INFINITY 33     //INT_MAX
#define MAX_VERTEX_NUM 20
#define InfoType int
#define VertexType char
typedef enum{ERROR,OK}Status;    //枚举型，函数状态变量


typedef struct ArcBox{                  //弧的结构
    int tailvex,headvex;                //该弧的弧头和弧尾
    struct ArcBox *hlink, *tlink;       //弧头相同及弧尾相同的弧的链域
    InfoType *info;                     //相关信息指针
}ArcBox;
typedef struct VexNode{                 //顶点的结构
    VertexType data;
    ArcBox *firstin, *firstout;         //分别指向该顶点的第一条入弧和第一条出弧
}VexNode;
typedef struct{
    VexNode xlist[MAX_VERTEX_NUM];      //表头向量
    int vexnum,arcnum;                  //顶点数目及弧的数目
}OLGraph;

//若G中存在顶点u，则返回u在G中的位置，若没找到则返回INFINITY
int LocateVex(OLGraph G, VertexType u){
    int i=0;
    for(i=0; i<G.vexnum; i++){
        if(G.xlist[i].data==u)
            break;
    }
    if(i<G.vexnum)
        return i;
    else
        return INFINITY;
}

//构建以十字链表为存储方式的有向图
Status CreateDG(OLGraph &G){
    char IncInfo;
    cout <<"Please input: vexnum(no more than 20) arcnum(no more than vexnum*vexnum) IncInfo(default 0)" << endl;
    cin >> G.vexnum >> G.arcnum >> IncInfo;
    cout << "构造顶点向量" << endl;
    for(int i=0; i<G.vexnum; i++){
        cin >> G.xlist[i].data;      //输入顶顶
        G.xlist[i].firstin=NULL;     //初始化指针
        G.xlist[i].firstout=NULL;
    }

    cout << "以'起点 终点'的方式依次输入每一条边(例如：ab\t起点：a，终点：b)：" << endl;
    VertexType v1,v2;
    int v1_int, v2_int;
    ArcBox *p;
    for(int k=0; k<G.arcnum; k++){
        cin >> v1 >> v2;
        v1_int = LocateVex(G, v1);
        v2_int = LocateVex(G, v2);
        p = new ArcBox;      //实例化一个弧结点
        //对结点赋值{tailvex, headvex, hlink, tlink, info}
        *p = {v1_int, v2_int, G.xlist[v2_int].firstin, G.xlist[v1_int].firstout, NULL};
        G.xlist[v2_int].firstin = p;
        G.xlist[v1_int].firstout = p;
            //if(IncInfo)
            //      input(*p->info)
    }
    return OK;
}

int main(){
    OLGraph G;
    CreateDG(G);

    for(int i=0; i<G.vexnum; i++)
        cout << G.xlist[i].data << "\t";
    cout << endl;

    ArcBox *p;
    for(int i=0; i<G.vexnum; i++){
        cout << G.xlist[i].data << " |\t";
        p=G.xlist[i].firstout;
        while(p){
            cout << G.xlist[p->tailvex].data << "_" << G.xlist[p->headvex].data << "    ";
```

```
84                  p=p->tlink;
85              }
86              cout << endl;
87          }
88
89
90          return 0;
91      }
92
```