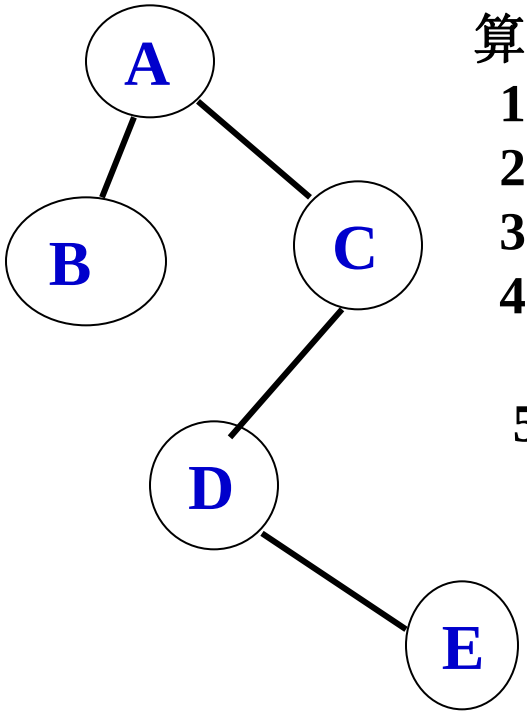
练习题：已知二叉树的逻辑结构如下，试写出建立二叉树的算法。

二叉树用三元组表示（data, parent, tag)，左图表示如下： (A,#,#),(B,A,L),(C,A,R),(D,C,L)
(E,D,R),(#,#,#)结束标志




算法思想：

1、建立一个空树；
2、用一个队列存放输入的点；
3、每输入一个结点，建立并入队；
4、若其parent为‘#’，则为根，否则到队列中查找父结点，没找到出队；
5、根据读入的tag与双亲建立关系
重复3—5。

```
typedef struct{
        char data, parent, tag ;
          } Bnode;
typedef struct{
          Btree ele[max];
           int front,rear;
           }Squeue;

Btree creat( )
{
  Bnode p;
  Squeue q;
  Btree  root, s;
```

```
typedef struct node{
    char data;
    node *lc,*rc}*Btree;
```

```
root=new node;
root=Null;
MakeNull(q);
cin>>p.data>>p.parent>>p.tag;
while(p.data!='#')
{
  s=new node;
  s->data=p.data;
  s->lc=s->rc=Null;
  q.rear=(q.rear+1)%max;
  q.ele[q.rear]=s; //入队
   if (p.parent=='#')
   root=s;
```

```
else//查找父结点
{
 while(!empty(q) && q.ele[q.front]->data!=p.parent)
       q.front=(q.fornt+1)%max;//出队
   if (q.ele [q.front]->data==p.parent)
        if(p.tag==‘L’)
             q.ele[q.front]->lc=s;
       else
             q.ele[q.front]->rc=s;
 }//else
   cin>>p.data>>p.parent>>p.tag;
   }//while
  return  root;
}
```


练习题：在二叉树中增加两个域parent父结点，flag标志，写出不用栈进行后序遍历的非递归算法

**flag**用于区分在遍历过程中达到该点时的走向（初始时每个结点 的**flag**均为**0**）.

```
struct node {
        char data;
        node *lc,*rc,*parent;
        int flag;
       };
```

```
struct node {       char data;       node *lc,*rc,*parent;
       int flag;
      };
void PostOrder ( node *t) //后序遍历二叉树/
{         node *p;
           p=t;   //t指向二叉树的根，建立时，所有结点标志为0/
           while( p!=Null)
       switch (p->flag)
        {              case 0: p->flag=1;
                               if (p- >lc!=Null) p=p->lc;
                               break;
           case 1:             p->flag=2;
                               if (p-> rc!=Null) p=p->rc;
                               break;
           case 2: p->flag=0;
                               cout<<p->data;
                               p=p->parent;
                               break;
         }
}
```

## 习题：

一棵二叉树的先序、中序和后序序列分别如下，其中一部分未显示出来，试求出空格处的内容，并画出该二叉树。

先序：_B_F_ICEH_G

中序：D_KFIA_EJC_

后序：_K_FBHJ_G_A

先序：ABDFKICEHJG

中序：DBKFIAHEJCG

后序：DKIFBHJEGCA