

```

void InorderTraversal( BinTree BT )
{
    if( BT ) {
        InorderTraversal( BT->Left );
        /* 此处假设对BT结点的访问就是打印数据 */
        printf("%d ", BT->Data); /* 假设数据为整型 */
        InorderTraversal( BT->Right );
    }
}

void PreorderTraversal( BinTree BT )
{
    if( BT ) {
        printf("%d ", BT->Data );
        PreorderTraversal( BT->Left );
        PreorderTraversal( BT->Right );
    }
}

void PostorderTraversal( BinTree BT )
{
    if( BT ) {
        PostorderTraversal( BT->Left );
        PostorderTraversal( BT->Right );
        printf("%d ", BT->Data);
    }
}

void LevelorderTraversal ( BinTree BT )
{
    Queue Q;
    BinTree T;

    if ( !BT ) return; /* 若是空树则直接返回 */

    Q = CreatQueue(); /* 创建空队列Q */
    AddQ( Q, BT );
    while ( !IsEmpty(Q) ) {
        T = DeleteQ( Q );
        printf("%d ", T->Data); /* 访问取出队列的结点 */
        if ( T->Left ) AddQ( Q, T->Left );
        if ( T->Right ) AddQ( Q, T->Right );
    }
}

```