

```

//
// Program to Demonstrate Tasks
//
#include <stdio.h>
#include <stdlib.h>
#include <omp.h>

struct tree {
    int data ;
    struct tree * left ;
    struct tree * right ;
};

typedef struct tree tree ;

#define NUM_NODES 16
#define DATA_MAX 100

void insert_data( int x, tree ** t )
{
    struct tree * u ;

    if (*t == NULL) {
        u = (tree *) malloc ( sizeof(tree) ) ;
        u->data = x ;
        u->left = NULL ;
        u->right = NULL ;
        *t = u ;
    }
    else {
        u = *t ;
        if ( u->data < x ) insert_data( x, &(u->left) ) ;
        else insert_data( x, &(u->right) ) ;
    }
}

tree * gen_tree( )
{
    int i, x ;
    tree * root ;

    root = NULL ;
    for( i = 0 ; i < NUM_NODES ; i++ ) {
        x = 1 + rand() % DATA_MAX ;
        insert_data( x, &root ) ;
    }
    return root ;
}

int delay()
{
    int i,x ;

    x = 0 ;
    for ( i = 0 ; i < 1000000000 ; i++ ) {
        if ( i & 0x1 ) x += i ;
        else x -= i ;
    }
    return x ;
}

```

```

/*-----*/
void postorder( tree * t )
{
    int id, dummy ;

    if ( t != NULL ) {

#pragma omp task firstprivate(t)
    {
        postorder( t->left ) ;
    }
#pragma omp task firstprivate(t)
    {
        postorder( t->right ) ;
    }

        id = omp_get_thread_num() ;
        printf( "(%d):  node = %lx ,  data = %d\n",
                id, ((unsigned long) t), t->data ) ;
        dummy = delay() ;
    }
}

/* ----- */
int main()
{
    tree * root ;

    root = gen_tree() ;
#pragma omp parallel firstprivate(root)
    {
        #pragma omp master
        {
            postorder( root ) ;
        }
    } /* End parallel region. */
}

```

===== Sample Session =====

```

cosmos% make
/bin/rm -f ptask *.o
gcc -c -fopenmp ptask.c
gcc -o ptask -fopenmp ptask.o
cosmos%
cosmos% setenv OMP_NUM_THREADS 4
cosmos% ptask
(0):  node = 1a3a010 ,  data = 84
(1):  node = 1a3a050 ,  data = 78
(3):  node = 1a3a030 ,  data = 87
(2):  node = 1a3a090 ,  data = 94
(0):  node = 1a3a0d0 ,  data = 87
(3):  node = 1a3a070 ,  data = 16
(2):  node = 1a3a0f0 ,  data = 93
(1):  node = 1a3a0b0 ,  data = 36
(0):  node = 1a3a190 ,  data = 91
(3):  node = 1a3a110 ,  data = 50
(2):  node = 1a3a130 ,  data = 22
(1):  node = 1a3a150 ,  data = 63
(3):  node = 1a3a170 ,  data = 28
(0):  node = 1a3a1d0 ,  data = 64
(2):  node = 1a3a1b0 ,  data = 60
(1):  node = 1a3a1f0 ,  data = 27

```