

Bài tập thực hành

Cấu trúc dữ liệu và thuật toán

Cây (1)

Bài 1: Viết chương trình:

1) Hoàn thiện khai báo cấu trúc dữ liệu của một nút của cây nhị phân:

```
typedef struct _bnode {  
    int item;  
    // Bổ sung mã chương trình nếu cần  
} BTreeNode;
```

2) Hoàn thiện hàm duyệt cây nhị phân theo thứ tự trước, giữa và sau.

```
void TreeTraversal_PreOrder(BTreeNode *cur) {  
    // Bổ sung mã chương trình nếu cần  
    printf("%d ", cur->item);  
    // Bổ sung mã chương trình nếu cần  
}
```

```
void TreeTraversal_InOrder(BTreeNode *cur) {  
    // Bổ sung mã chương trình nếu cần  
    printf("%d ", cur->item);  
    // Bổ sung mã chương trình nếu cần  
}
```

```
void TreeTraversal_PostOrder(BTreeNode *cur) {  
    // Bổ sung mã chương trình nếu cần  
    printf("%d ", cur->item);  
    // Bổ sung mã chương trình nếu cần  
}
```

3) Hoàn thiện hàm main()

```
int main(void) {  
    int item1 = 1, item2 = 2, item3 = 3;  
    BTreeNode bnode1, bnode2, bnode3;  
    bnode1.item = item1;  
    bnode2.item = item2;  
    bnode3.item = item3;  
  
    bnode2.left = NULL;  
    bnode2.right = NULL;  
  
    bnode3.left = NULL;  
    bnode3.right = NULL;
```

```

    btnode1.left = ____ ;
    btnode1.right = ____ ;

    printf("\nDuyet cay theo thu tu truoc: ");
    TreeTraversal_PreOrder( ____ );

    printf("\nDuyet cay theo thu tu giua: ");
    TreeTraversal_InOrder( ____ );

    printf("\nDuyet cay theo thu tu sau: ");
    TreeTraversal_PostOrder( ____ );
}

```

Bài 2: Cho biết kết quả khi duyệt cây theo thứ tự trước, giữa, sau:

