≡ PTA

ZJU_FDS_MidTermExam

判断题 8　A. 单选题 13　程序填空题 2

5-1 Concatenation of lists is an operation where the elements of one list are added at the end of another list. For example, if we have a linked list `L1`→1→2→3 and another one `L2`→4→5→6. The function `ListConcat` is to return the head pointer of the list L→4→5→6→1→2→3.

作者	陈越
单位	浙江大学
时间限制	400 ms
内存限制	64 MB

The list structure is defined as the following:

```
typedef struct Node *PtrToNode;
struct Node{
    int Data;
    PtrToNode Next;
};
typedef PtrToNode List;
```

Please fill in the blanks.

```
List ListConcat( List L1, List L2 )
{
    List Tmp = L2;
    if ( !L2 ) return L1;
    while ( Tmp->Next )
        Tmp=Tmp->Next (3分);
    Tmp->Next=L1->Next (3分);
    return L2 (3分);
}
```

5-1 部分正确 (6 分)

5-2 The function `BuildTree` is to build and return a binary tree from its inorder and preorder traversal sequences.

作者	陈越
单位	浙江大学
时间限制	400 ms
内存限制	64 MB

The tree structure is defined as the following:

```
typedef struct Node *PtrToNode;
struct Node{
    int Data;
    PtrToNode Left, Right;
};
typedef PtrToNode Tree;
```

Please fill in the blanks.

```
Tree BuildTree( int in[], int pre[], int N )
{ //in[] stores the inorder traversal sequence
  //and pre[] stores the preorder traversal sequence
  //N is the number of nodes in the tree
     Tree T;
     int i;

     if (!N) {
        return NULL;
     }
     T = (Tree)malloc(sizeof(struct Node));
     T->Data = pre[0] (3分);
     for (i=0; i<N; i++)
         if (in[i]==T->Data) break;
     T->Left = BuildTree( in, pre+1, i (3分));
     T->Right = BuildTree( in+i+1, pre+i+1, N-i-1 (3分));
     return T;
}
```

5-2 答案正确 (9 分)