Q11

E14086020 洪緯宸

講解binary tree insertion work 實作

將資料讀入

n 為input數量

r 為root id

```
int n, r, insert_node, insert_element;
char command[6];
Node *cur = (Node *)malloc(sizeof(struct Node));
scanf("%d%d", &n, &r);
root = newNode(r);
```

再來把以下資料一個一個插進去我們的binary tree

首先用find_node找到要插入的node 並回傳node地址

然後讀入照著command 插入左邊或右邊

```
for (int i = 0; i < n; ++i)
{
    scanf("%d%s%d", &insert_node, command, &insert_element);
    cur = find_node(insert_node);
    if (command[0] == 'r')
    {
        insertRight(cur, insert_element);
    }
    else
    {
        insertLeft(cur, insert_element);
    }
}
```

find_node 實作

用iterative preorder的方式 搜尋binary tree裡的所有node 如果等於val 就回傳此node的地址

```
Node *find_node(int val)
{
    int top = 0;
    Node *stack[MAX_N];
    Node *cur = (Node *)malloc(sizeof(struct Node));
    stack[++top] = root;
    while (top)
    {
        cur = stack[top--];
        if (cur->val == val)
            return cur;
        if (cur->right != NULL)
            stack[++top] = cur->right;
        if (cur->left != NULL)
            stack[++top] = cur->left;
    }
    return NULL;
}
```

插入right node 跟 left node 實作

```
void insertLeft(Node *node, int left_val)
{
    node->left = newNode(left_val);
}

void insertRight(Node *node, int right_val)
{
    node->right = newNode(right_val);
}
```

new_node 實作

malloc 一個空間再回傳此地址

```
Node *newNode(int val)
{
    Node *node = (struct Node *)malloc(sizeof(struct Node));
    node->val = val;
    node->left = NULL;
    node->right = NULL;
    return node;
}
```

輸出結果

根據題目要求輸出inorder traversal

```
void printInorder(Node *node)
{
    if (node == NULL)
        return;
    printInorder(node->left);
    printf("%d ", node->val);
    printInorder(node->right);
}
```