/\*

240508 트리

\*/

1. **스레드 이진 트리**

#include <stdio.h>

#define \_CRT\_SECURE\_NO\_WARNINGS

#pragma warning(disable:4996)

#define TRUE 1

#define FALSE 0

typedef struct TreeNode {

int data;

struct TreeNode\* left, \* right;

int is\_thread;// 스레드이면 TRUE

} TreeNode;

// G

// C F

// A B D E

TreeNode n1 = { 'A', NULL, NULL, 1 };

TreeNode n2 = { 'B', NULL, NULL, 1 };

TreeNode n3 = { 'C', &n1, &n2, 0 };

TreeNode n4 = { 'D', NULL, NULL, 1 };

TreeNode n5 = { 'E', NULL, NULL, 0 };

TreeNode n6 = { 'F', &n4, &n5, 0 };

TreeNode n7 = { 'G', &n3, &n6, 0 };

TreeNode\* exp = &n7;

TreeNode\* find\_successor(TreeNode\* p)

{

// q는 p의 오른쪽 포인터

TreeNode\* q = p->right;

// 만약 오른쪽 포인터가 NULL이거나 스레드이면 오른쪽 포인터를 반환

if (q == NULL || p->is\_thread == TRUE)

return q;

// 만약 오른쪽 자식이면 다시 가장 왼쪽 노드로 이동

while (q->left != NULL) q = q->left;

return q;

}

void thread\_inorder(TreeNode\* t)

{

TreeNode\* q;

q = t;

while (q->left) q = q->left;// 가장 왼쪽 노드로 간다.

do {

printf("%c -> ", q->data);// 데이터 출력

q = find\_successor(q); // 후속자 함수 호출

} while (q);// NULL이 아니면

}

int main(void)

{

// 스레드 설정

n1.right = &n3;

n2.right = &n7;

n4.right = &n6;

// 중위 순회

thread\_inorder(exp);

}