

```
node_pi->siblingLeft->siblingRight = NULL;
free(node_pi);
node_ri = node_pi = node_rn = node_ty = NULL;
```

구현 상 생기는 마지막 잉여 노드를 없애는 코드

하지만 총 노드가 0개라면 node_pi->siblingLeft 값이 존재하지 않기 때문에 node_pi->siblingLeft->siblingRight 연산에서 Segmentation fault 발생!

```
if(node_pi->siblingLeft) node_pi->siblingLeft->siblingRight = NULL;
else head = NULL;
Free_Node(node_pi);
node_ri = node_pi = node_rn = node_ty = NULL;
```

```
if(node_pi->siblingLeft) node_pi->siblingLeft->siblingRight = NULL;
else head = NULL;
Free_Node(node_pi);
node_ri = node_pi = node_rn = node_ty = NULL;
```

이걸로 바꿔주면 됨!

```
void Free_Node(Node *node) {
    free(node->ri);
    free(node->rn);
    free(node->pi);
    free(node);
}
```

요건 Free_Node 함수!

```
node_pi->siblingRight = (Node*)malloc(sizeof(Node));
node_pi->siblingRight->siblingLeft = node_pi;
node_pi = node_pi->siblingRight;
```

요건 새로운 Node 할당하는 부분인데 malloc으로만 동적 할당하면 Node 내에 변수들이 이상한 값들이 들어가서 트리 순회할 때 오류가 발생함!

```
node_pi->siblingRight = Create_Node("", "", "", 0);
node_pi->siblingRight->siblingLeft = node_pi;
node_pi = node_pi->siblingRight;
```

동적 할당 후 초기화까지 진행해주는 별도 함수로 노드를 할당해주는 코드로 변경

```

Node* Create_Node(char *ri, char *rn, char *pi, ObjectType ty){
    Node* node = (Node*)malloc(sizeof(Node));

    node->rn = (char*)malloc(sizeof(rn));
    node->ri = (char*)malloc(sizeof(ri));
    node->pi = (char*)malloc(sizeof(pi));

    strcpy(node->rn, rn);
    strcpy(node->ri, ri);
    strcpy(node->pi, pi);

    node->parent = NULL;
    node->child = NULL;
    node->siblingLeft = NULL;
    node->siblingRight = NULL;
    node->ty = ty;

    if(strcmp(rn,"") && strcmp(rn,"TinyIoT")) {
        fprintf(stderr,"\nCreate Tree Node\n[rn] %s\n[ri] %s\n",node->rn,node->ri);
    }

    return node;
}

```

요건 Create_Node 함수!