AppData\Local\Temp\d2598f46-e461-4b38-9fa1-0b534a174477_OSAP_003_7_최종보고서(소스코드 포함).zip.477\src\ancestor.cc

```
/*
MIT License
This file is part of the INHA_OSAP_003_7 project.
Copyright (c) 2024 tbmyong

Permission is hereby granted, free of charge, to any person obtaining a copy
of this software and associated documentation files (the "Software"), to deal
in the Software without restriction, including without limitation the rights
to use, copy, modify, merge, publish, distribute, sublicense, and/or sell
copies of the Software, and to permit persons to whom the Software is
furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all
copies or substantial portions of the Software.

THE SOFTWARE IS PRPpOVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR
IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY,
FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE
AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER
LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM,
OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE
SOFTWARE.

작성자: 임시영
작성일(파일 생성일): 2024-12-05
작성일(파일 최종 수정일): 2024-12-15
*/

/*
<Ancestor 기능 구현>
Ancestor()
: key 값을 지닌 노드를 찾아서 해당 노드의 깊이와 높이를 더한 값과
  조상 노드의 key 값의 합 계산한다. 트리의 루트 노드부터 탐색하고
  경로 상의 부모 노드 key 값을 누적해 depth_height_sum 계산한다.
*/

#include "../base/avl.h"

// 조상 노드들의 합을 계산하는 함수
int AVL::CalculateAncestorSum(Node* root, int key) {
  int ancestor_sum = 0;
  Node* current_node = root;

  while (current_node != nullptr) {
    // 키를 찾으면 합산 완료
    if (key == current_node->get_key()) {
      return ancestor_sum;
    }

    ancestor_sum += current_node->get_key();

```

```
52      current_node = (key < current_node->get_key()) ? current_node->get_left()
53                                                     : current_node->get_right();
54    }
55
56    return 0;
57  }
58
59  std::pair<int, int> AVL::Ancestor(int key) {
60    int ancestor_sum = CalculateAncestorSum(get_root(), key);
61    int depth_height_sum = CalculateDepthHeightSum(get_root(), key);
62
63    // key값 찾으면 결과 반환
64    if (depth_height_sum != 0) {
65      return {depth_height_sum, ancestor_sum};
66    }
67
68    return {0, 0};
69  }
70
```