Problem Solving (DS14)

هذا البرنامج التدريبي مُصاغ بعناية لتمكين المتدربين من تطوير قدراتهم الفكرية على غرار المبرمجين المحترفين، والتعاون بكفاءة ضمن فريق محترف في شركة "جمال تك" أو أي مؤسسة متعددة الجنسيات أخرى. نظرًا لأهمية اللغة الإنجليزية في بيئة العمل العالمية، يتم تقديم المحتوى التدريبي بالإنجليزية. لا يشترط إتقان اللغة بشكل كامل، لكن من الضروري امتلاك القدرة الكافية لفهم المتطلبات وتنفيذها بشكل فعّال. يُمكن للمتدربين استخدام مترجم جوجل أو الاستعانة بـ "شات جي بي تي" للتغلب على أية عقبات لغوية، المهم هو الفهم الدقيق للمطلوب وتحقيقه بنجاح.

لتعظيم الاستفادة من التدريب، يُنصح بمحاولة حل التمارين بشكل مستقل لمدة ساعة واحدة على الأقل قبل الرجوع إلى الحل المرفق في نهاية الملف.

قد يتضمن الحل كودًا برمجيًا غير مفسر بعد، والغرض من ذلك هو تشجيعك على محاولة فهم الأكواد البرمجية الجديدة التي لم تتعرض لها من قبل. هذه المهارة ضرورية في سوق العمل، حيث تتطور لغات البرمجة باستمرار ويظهر كل يوم لغات جديدة. ستواجه دائمًا أكوادًا لم تدرسها من قبل، ومن المهم أن تكون قادرًا على فهمها بنفسك دون الحاجة إلى دراسة مسبقة. يمكنك الاستعانة بمحرك البحث جوجل، أو استخدام ChatGPT، أو حتى اللجوء لأصدقائك للمساعدة. الهدف الأساسي هو أن تصل إلى فهم معنى كل كود بأي طريقة ممكنة لتتمكن من إيجاد موقعك في سوق العمل.

إن وجود كود برمجي غير مفسر يشكل تحديًا يتوجب عليك إيجاد حل له. هذا النوع من التدريبات يعد جزءًا أساسيًا من تدريبات 'Problem Solving'، التي تهدف إلى تمكينك من أداء عملك بفاعلية بغض النظر عن التحديات والعقبات. هذه القدرة على حل المشكلات هي ما يتمتع به العاملون في 'جمال تك'، ومن الضروري أن تطور في نفسك هذه المهارة لتصبح عضوًا فعالًا في فريق عمل 'جمال تك'.

Gammal Tech's Linked List Challenge

Background

Gammal Tech, a top-tier software development company, renowned for its innovation and state-of-the-art technology, is working on a revolutionary project. The project involves optimizing data flow within a highly complex network system. To tackle this, they turn to their tried and true data structures, with a special focus on linked lists. However, this isn't just any linked list; it's a linked list with a twist.

Problem Statement

Gammal Tech's network system is represented as a linked list of nodes, where each node contains a unique data value and a pointer to the next node. However, in this special linked list, each node also has a 'jump' pointer to any other node in the list (including itself). Gammal Tech's task is to write a program that calculates the sum of

the data values of the nodes that are reached by following the 'jump' pointers, starting from the head node and ending when a node points to itself.

Constraints

The number of nodes in the linked list, `N`, is between 1 and 10^5.
Each data value in the node is an integer between 1 and 10^6.
The list contains no cycles, except through the 'jump' pointers.

Input Format

- The first line contains a single integer `N`, the number of nodes in the linked list.
- The next `N` lines each contain two integers. The first integer is the data value of the node, and the second integer is the index of the node to which the 'jump' pointer points (0-indexed). If a node points to itself, the second integer is its own index.

Output Format

Print a single integer - the sum of the data values of the nodes reached by following the 'jump' pointers.

Sample Input

```
5
10 1
20 2
15 4
25 3
30 4
```

Sample Output

```
75
```

Explanation: The 'jump' pointers form the following sequence: 10 -> 20 -> 15 -> 30. The node with value 30 points to itself, so the sum is 10 + 20 + 15 + 30 = 75.

لتعظيم الاستفادة من التدريب، يُنصح بمحاولة حل التمرين بشكل مستقل لمدة ساعة واحدة على الأقل قبل الرجوع إلى الحل المرفق

C++ Solution:

```cpp
#include <iostream>
#include <vector>
using namespace std;

int main() {
    int N;
    cin >> N;

    vector<pair<int, int>> nodes(N);
    for (int i = 0; i < N; ++i) {
        cin >> nodes[i].first >> nodes[i].second;
    }

    int sum = 0, currentIndex = 0;
    while (true) {
        sum += nodes[currentIndex].first;
        if (nodes[currentIndex].second == currentIndex) break;
        currentIndex = nodes[currentIndex].second;
    }

    cout << sum << endl;
    return 0;
}
```