



## Problem Solving (DS16)

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 فعال. 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 Green Effort

### Background

Gammal Tech, a renowned software development company, is known for its innovative approach to problem-solving and dedication to environmental preservation. In a new initiative, the company aims to optimize its resource allocation in a way that minimizes environmental impact.

### Problem Statement

Gammal Tech has several ongoing projects, each consuming a varying amount of resources. The company has decided to use a linked list to track these projects due to its dynamic nature. Each node in the linked list represents a project, holding the project ID and the amount of resources it consumes.



Gammal Tech seeks to develop a software tool that calculates the total resource consumption in an efficient manner. The twist is to implement this calculation using recursion, showcasing the company's innovative approach.

### Constraints

- $1 \leq N \leq 1000$  (where  $N$  is the number of projects)
- $1 \leq \text{Project ID} \leq 10^5$
- $1 \leq \text{Resources consumed by each project} \leq 10^3$

### Input Format

- The first line of input contains an integer  $N$ , the number of projects.
- The next  $N$  lines each contain two integers, the Project ID and the amount of resources it consumes.

### Output Format

- Output a single integer, the total amount of resources consumed by all projects.

### Sample Input

```
5
101 50
102 40
103 60
104 30
105 20
```

### Sample Output

```
200
```

### Explanation

The total resources consumed are  $50 + 40 + 60 + 30 + 20 = 200$ .

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



## C++ Solution:

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

struct Node {
    int projectID;
    int resources;
    Node* next;
};

// Function to add a new node to the linked list
void insertNode(Node** head_ref, int projectID, int resources) {
    Node* new_node = new Node();
    new_node->projectID = projectID;
    new_node->resources = resources;
    new_node->next = (*head_ref);
    (*head_ref) = new_node;
}

// Recursive function to calculate total resources
int totalResources(Node* node) {
    if (node == NULL)
        return 0;
    return node->resources + totalResources(node->next);
}

// Main function
int main() {
    Node* head = NULL;
    int n, projectID, resources;

    cin >> n;
    for(int i = 0; i < n; i++) {
        cin >> projectID >> resources;
        insertNode(&head, projectID, resources);
    }

    cout << totalResources(head) << endl;
    return 0;
}
```