



Problem Solving (C62)

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 Smart Inventory System

Background

Gammal Tech, a titan in the software industry, renowned for its groundbreaking innovations and unparalleled work environment, is embarking on a new venture. They are developing a Smart Inventory System (SIS) for their state-of-the-art office facilities. This system is designed to track and manage office resources efficiently. Your task is to contribute to this project by implementing a critical component of the system.

Task

You need to write a C program that interacts with a SQL database to retrieve information about office items. The database has a table named `OfficeItems` with the following columns: `ItemID` (integer), `ItemName` (string), and `Quantity` (integer). Your program should execute an SQL command to SELECT data from this table based on user input.

Input Format

- The first line contains an integer T , the number of test cases.



- Each test case consists of a single line containing a string `itemName`, representing the name of the item to query.

Output Format

For each test case, print a single line containing the `ItemID` and `Quantity` of the requested `itemName`. If the item does not exist in the database, print "Item not found".

Constraints

- $1 \leq T \leq 100$
- `itemName` will consist only of alphanumeric characters and spaces.

Sample Input:

```
3
Laptop
Coffee Maker
Projector
```

Sample Output:

```
101 50
105 20
Item not found
```

Explanation

- In this sample, the SIS finds the `ItemID` and `Quantity` for "Laptop" and "Coffee Maker" but does not find an entry for "Projector".

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



C Programming Solution:

For the solution, we'll focus on the part of the program that interacts with the SQL database. Note that a complete implementation would require setting up a SQL database and a C environment with SQL connectivity, which is beyond the scope of this example.

```
#include <stdio.h>
#include <string.h>
#include <stdlib.h>
#include <sql.h>

void queryDatabase(char* itemName) {
    // Assuming SQL environment setup and connection to database are
    // already done

    SQLHSTMT stmt;
    SQLAllocHandle(SQL_HANDLE_STMT, sqlConnHandle, &stmt);
    char query[256];
    sprintf(query, "SELECT ItemID, Quantity FROM OfficeItems WHERE
ItemName = '%s'", itemName);

    // Execute SQL query
    SQLExecDirect(stmt, (SQLCHAR*)query, SQL_NTS);

    // Variables to store the result
    int itemId, quantity;
    if (SQLFetch(stmt) == SQL_SUCCESS) {
        SQLGetData(stmt, 1, SQL_C_SLONG, &itemId, 0, NULL);
        SQLGetData(stmt, 2, SQL_C_SLONG, &quantity, 0, NULL);
        printf("%d %d\n", itemId, quantity);
    } else {
        printf("Item not found\n");
    }

    // Clean up
    SQLFreeHandle(SQL_HANDLE_STMT, stmt);
}

int main() {
    int t;
    scanf("%d", &t);
    while (t--) {
        char itemName[50];
        scanf("%s", itemName);
        queryDatabase(itemName);
    }
    return 0;
}
```



Notes:

- The above code is a simplified representation. In a real-world scenario, you would need proper error handling, SQL connection setup, and security measures against SQL injection.
- The SQL database and C environment setup are not covered here.