



### Problem Solving (C23)

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. 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 3D Space Mapper

### Background:

At Gammal Tech, creativity and the ability to imagine complex structures are highly valued traits. To cultivate these skills, especially in handling multidimensional arrays, Gammal Tech introduces an enhanced exercise. This task involves a 3D array, representing a three-dimensional space, where learners can input the starting number for each layer, fostering a deeper understanding and interaction with multi-layered data structures.

### Problem Statement:

The challenge is to write a C program that creates a 3x3x3 multidimensional array and allows the user to specify the starting number for each layer. The program should then populate each layer with consecutive numbers starting from the given initial value and display the array, visualizing the 3D structure.



This exercise aims to help beginners visualize and manipulate data within a 3D array, simulating a simplified version of the complex data management found in Gammal Tech's software systems.

### Your Task:

Write a C program that:

1. Prompts the user to enter the starting number for each of the three layers in a 3x3x3 array.
2. Initializes the array based on these starting numbers.
3. Displays the array in a format that represents the 3D structure.

### Sample Input:

```
Enter the starting number for layer 1: 1
Enter the starting number for layer 2: 10
Enter the starting number for layer 3: 20
```

### Sample Output:

```
Layer 1:
1 2 3
4 5 6
7 8 9

Layer 2:
10 11 12
13 14 15
16 17 18

Layer 3:
20 21 22
23 24 25
26 27 28
```

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



## C Programming Solution:

```
#include <stdio.h>

#define SIZE 3

int main() {
    int space[SIZE][SIZE][SIZE], startNum;

    // Prompt and initialize each layer with user-provided starting
    numbers
    for (int layer = 0; layer < SIZE; layer++) {
        printf("Enter the starting number for layer %d: ", layer + 1);
        scanf("%d", &startNum);

        for (int row = 0; row < SIZE; row++) {
            for (int col = 0; col < SIZE; col++) {
                space[layer][row][col] = startNum++;
            }
        }
    }

    // Display the 3D array
    for (int i = 0; i < SIZE; i++) {
        printf("Layer %d:\n", i + 1);
        for (int j = 0; j < SIZE; j++) {
            for (int k = 0; k < SIZE; k++) {
                printf("%d ", space[i][j][k]);
            }
            printf("\n");
        }
        printf("\n");
    }

    return 0;
}
```