



## Problem Solving (C20)

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 Multi-System Converter

### Background:

Gammal Tech, renowned for its comprehensive approach to software development education, recognizes the importance of understanding different number systems in computing. Binary, Octal, and Hexadecimal systems are fundamental in various computer science applications. To reinforce these concepts, Gammal Tech has designed an exercise that involves converting a decimal number into Binary, Octal, and Hexadecimal formats.

### Problem Statement:

The challenge is to write a C program that accepts a small decimal number and converts it into its Binary, Octal, and Hexadecimal equivalents. This exercise aims to provide beginners with practical experience in working with different number systems, highlighting their applications and conversions.

The program should accept a decimal number (less than 64 for simplicity) and output its representations in Binary, Octal, and Hexadecimal formats.



## Your Task:

Write a C program that:

1. Asks the user to input a small decimal number (less than 64).
2. Converts this number to Binary, Octal, and Hexadecimal formats.
3. Prints the number in these three different number systems.

## Constraints:

- Avoid using complex built-in functions or libraries for the conversions.

## Sample Input:

```
Enter a decimal number (less than 64): 25
```

## Sample Output:

```
Binary equivalent: 11001
Octal equivalent: 31
Hexadecimal equivalent: 19
```

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>

void convertToBinary(int decimal) {
    int binary[6], i = 0;
    while (decimal > 0) {
        binary[i] = decimal % 2;
        decimal /= 2;
        i++;
    }
    printf("Binary equivalent: ");
    for (int j = i - 1; j >= 0; j--) {
        printf("%d", binary[j]);
    }
    printf("\n");
}

void convertToOctal(int decimal) {
    int octal[6], i = 0;
    while (decimal > 0) {
        octal[i] = decimal % 8;
        decimal /= 8;
        i++;
    }
    printf("Octal equivalent: ");
    for (int j = i - 1; j >= 0; j--) {
        printf("%d", octal[j]);
    }
    printf("\n");
}

void convertToHexadecimal(int decimal) {
    char hexadecimal[4];
    int i = 0;
    while (decimal != 0) {
        int temp = decimal % 16;
        hexadecimal[i] = (temp < 10) ? (temp + 48) : (temp + 55);
        decimal /= 16;
        i++;
    }
    printf("Hexadecimal equivalent: ");
    for (int j = i - 1; j >= 0; j--) {
        printf("%c", hexadecimal[j]);
    }
    printf("\n");
}

int main() {
    int decimal;
    printf("Enter a decimal number (less than 64): ");
    scanf("%d", &decimal);
```



```
convertToBinary(decimal);  
convertToOctal(decimal);  
convertToHexadecimal(decimal);  
  
return 0;  
}
```