

سلسلة الخوارزميات وحل المشاكل - المستوى الثاني



26+ Years
of Experience

PROGRAMMING ADVICES

LEARN THE
RIGHT WAY

Mohammed Abu-Hadhoud

MBA, PMOC, PgMP®, PMP®, PMI-RMP®, CM, ITIL®, MCPD, MCSD



حقوق النشر محفوظة، أسعار الكورسات في المنصة هي أسعار
رمزية جدا، ارجو عدم نشر هذه الوثيقة لان نشرها سيمنعنا من
الاستمرار في تقديم العلم للآخرين

ارجو عدم استخدام هذه الوثيقة من غير وجه حق لأنك ستحرم الاف
الناس من التعلم

ProgrammingAdVICES.com



Problem # 40/2 Solution Using C++

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

void FillArray(int arr[100], int& arrLength)
{
    arrLength = 10;
    arr[0] = 10;
    arr[1] = 10;
    arr[2] = 10;
    arr[3] = 50;
    arr[4] = 50;
    arr[5] = 70;
    arr[6] = 70;
    arr[7] = 70;
    arr[8] = 70;
    arr[9] = 90;
}

void PrintArray(int arr[100], int arrLength)
{
    for (int i = 0; i < arrLength; i++)
        cout << arr[i] << " ";

    cout << "\n";
}

short FindNumberPositionInArray(int Number, int arr[100], int
arrLength)
{
    for (int i = 0; i < arrLength; i++)
    {
        if (arr[i] == Number)
            return i; //return the index
    }

    //if you reached here, this means the number is not found
    return -1;
}

bool IsNumberInArray(int Number, int arr[100], int arrLength)
{
    return FindNumberPositionInArray(Number, arr, arrLength) != -1;
}
```



Problem # 40/2 Solution Using C++

```
void AddArrayElement(int Number, int arr[100], int& arrLength)
{
    //its a new element so we need to add the length by 1
    arrLength++;
    arr[arrLength - 1] = Number;
}

void CopyDistinctNumbersToArray(int arrSource[100], int
arrDestination[100], int SourceLength, int& DestinationLength)
{
    for (int i = 0; i < SourceLength; i++)
    {
        if (!IsNumberInArray(arrSource[i], arrDestination,
DestinationLength))
        {
            AddArrayElement(arrSource[i], arrDestination,
DestinationLength);
        }
    };
}

int main()
{
    int arrSource[100], SourceLength = 0, arrDestination[100],
DestinationLength = 0;
    FillArray(arrSource, SourceLength);

    cout << "\nArray 1 elements:\n";
    PrintArray(arrSource, SourceLength);

    CopyDistinctNumbersToArray(arrSource, arrDestination,
SourceLength, DestinationLength);

    cout << "\nArray 2 distinct elements:\n";
    PrintArray(arrDestination, DestinationLength);

    return 0;
}
```