

# Count and Print Duplicate Numbers

## English Version

### Problem Statement

You are given an array of integers. Your task is to:

1. Count how many distinct numbers appear **more than once** in the array.
2. Print each of these **duplicate numbers** in the **order of their first appearance**.

### Input Format

- The first line contains a single integer  $n$  ( $1 \leq n \leq 100$ ), the number of elements in the array.
- The second line contains  $n$  space-separated integers  $a[i]$  ( $0 \leq a[i] < 10000$ ).

### Output Format

- First, print a single integer — the count of distinct numbers that appear more than once.
- Then, on the same line, print all such numbers in the order of their **first appearance**, separated by a space.

### Example Input

```
10
1 2 3 2 1 4 5 1 3 3
```

### Example Output

```
3 1 2 3
```

### Explanation

- $1$  appears 3 times
- $2$  appears 2 times
- $3$  appears 3 times
- All of these are duplicates, and they are printed in the order they first appear in the array.

---

## Vietnamese Version (Phiên bản tiếng Việt)

## Đề bài

Bạn được cho một mảng các số nguyên. Nhiệm vụ của bạn là:

- Đếm có bao nhiêu số khác nhau xuất hiện **nhều hơn một lần** trong mảng.
- In ra từng **số trùng lặp** này theo **thứ tự xuất hiện đầu tiên** của chúng.

## Định dạng đầu vào

- Dòng đầu tiên chứa một số nguyên  $n$  ( $1 \leq n \leq 100$ ), là số phần tử trong mảng.
- Dòng thứ hai chứa  $n$  số nguyên  $a[i]$  cách nhau bằng dấu cách ( $0 \leq a[i] < 10000$ ).

## Định dạng đầu ra

- Đầu tiên, in ra một số nguyên — số lượng các số khác nhau xuất hiện nhiều hơn một lần.
- Sau đó, trên cùng một dòng, in ra tất cả các số đó theo thứ tự **xuất hiện đầu tiên**, cách nhau bằng dấu cách.

## Ví dụ đầu vào

```
10
1 2 3 2 1 4 5 1 3 3
```

## Ví dụ đầu ra

```
3 1 2 3
```

## Giải thích

- Số 1 xuất hiện 3 lần
- Số 2 xuất hiện 2 lần
- Số 3 xuất hiện 3 lần
- Tất cả các số này đều là trùng lặp, và chúng được in theo thứ tự xuất hiện đầu tiên trong mảng.

## Sample C Code Implementation

```
c
```

```

#include <stdio.h>
int main()
{
    int n, a[101];
    scanf("%d", &n);
    for (int i = 0; i < n; i++)
        scanf("%d", &a[i]);
    int f[10000] = {0};
    for (int i = 0; i < n; i++)
    {
        f[a[i]]++;
    }
    int b = 0;
    for (int i = 0; i < 10000; i++)
    {
        if (f[i] > 1)
            b++;
    }
    printf("%d ", b);
    for (int i = 0; i < n; i++)
    {
        if (f[a[i]] > 1)
            printf("%d ", a[i]);
        f[a[i]] = 0;
    }
    return 0;
}

```

## Algorithm Approach (Phương pháp thuật toán)

### Code Explanation:

1. **Read input:** Read  $n$  and the array elements
2. **Frequency counting:** Use array `f[10000]` to count occurrences of each number
3. **Count duplicates:** Count how many numbers have frequency  $> 1$
4. **Print results:** Output count and duplicate numbers in order of first appearance
5. **Mark as processed:** Set frequency to 0 after printing to avoid duplicates

### Steps (Các bước):

1. **Track first occurrence:** Use a dictionary/map to store the first index where each number appears

2. **Count frequencies:** Count how many times each number appears
3. **Filter duplicates:** Identify numbers that appear more than once
4. **Sort by first occurrence:** Sort duplicate numbers by their first appearance order
5. **Output results:** Print count and the sorted duplicate numbers

#### **Các bước (Vietnamese):**

1. **Theo dõi lần xuất hiện đầu tiên:** Sử dụng từ điển/bản đồ để lưu chỉ số đầu tiên mà mỗi số xuất hiện
2. **Đếm tần suất:** Đếm số lần mỗi số xuất hiện
3. **Lọc các số trùng lặp:** Xác định các số xuất hiện nhiều hơn một lần
4. **Sắp xếp theo thứ tự xuất hiện đầu tiên:** Sắp xếp các số trùng lặp theo thứ tự xuất hiện đầu tiên
5. **Xuất kết quả:** In ra số lượng và các số trùng lặp đã sắp xếp