Reversing an Array in C

Problem Statement

Given an array of size n, reverse it.

Example: If the array is `[16, 13, 7, 2, 1, 12]`, after reversing, the array should be `[12, 1, 2, 7, 13, 16]`.

Input Format

- The first line contains an integer n, denoting the size of the array.
- The next line contains n space-separated integers denoting the elements of the array.

Constraints

$$1 \leq n \leq 10^5, \quad andeachelementa_iisaninteger.$$

Output Format

Print the reversed array elements in a single line separated by spaces.

Sample Input 0

```
6
16 13 7 2 1 12
```

Sample Output 0

```
12 1 2 7 13 16
```

Explanation

The original array is `[16, 13, 7, 2, 1, 12]`. After reversing, the array becomes `[12, 1, 2, 7, 13, 16]`.

Solution in C

```c
// Program to reverse an array

#include <stdio.h>

int main() {
    int n;
    scanf("%d", &n);

    int arr[n];
    for(int i = 0; i < n; i++) {
        scanf("%d", &arr[i]);
    }

    // Reverse the array
    for(int i = 0; i < n / 2; i++) {
        int temp = arr[i];
        arr[i] = arr[n - 1 - i];
        arr[n - 1 - i] = temp;
    }

    // Print the reversed array
    for(int i = 0; i < n; i++) {
        printf("%d ", arr[i]);
    }

    return 0;
}
```

Common Problem Ideas Related to Array Reversal

- **Rotate the Array:** Instead of reversing, rotate the array by k positions to the left or right.
- **Reverse Subarrays:** Reverse only a specific portion of the array, given start and end indices.
- **Check Palindrome Array:** Check if an array reads the same forward and backward.
- **Merge Two Reversed Arrays:** Reverse two arrays separately and then merge them.
- **Reverse Using Recursion:** Implement array reversal using recursion instead of loops.
- **Reverse with Additional Data Structures:** Use a stack or queue to reverse the array.

How to Make the Problem Harder

- **Memory Constraints:** Reverse very large arrays where extra memory allocation is restricted.
- **In-Place Reversal without Temporary Variables:** Reverse the array without using an explicit temporary variable for swapping (e.g., using XOR swapping).
- **Multidimensional Arrays:** Reverse rows or columns of 2D arrays or reverse the entire 2D array.
- **Linked List Reversal:** Extend the problem to reversing linked lists or other data structures.
- **Streaming Data:** Reverse elements as they come in a stream without storing the entire array at once.
- **Parallel Reversal:** Implement array reversal using parallel or multi-threaded approaches for performance.