GCD array in C++

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

// Function to compute GCD of two numbers using Euclidean algorithm
int gcd(int a, int b) {
    while (b != 0) {
        int temp = b;
        b = a % b;
        a = temp;
    }
    return a;
}

// Function to compute GCD of an array of integers
int gcdArray(vector<int>& arr) {
    int result = arr[0];
    for (int i = 1; i < arr.size(); i++) {
        result = gcd(result, arr[i]);
        if (result == 1) { // If result becomes 1, further GCD will also be 1
            return 1;
        }
    }
    return result;
}

int main() {
    vector<int> arr = {12, 24, 36, 48};
    cout << "GCD of the array elements: " << gcdArray(arr) << endl;
    return 0;
}
```

Step-by-Step Dry Run (Tabular Form)

We'll use this table to track the intermediate GCD results:

Step	`result` (previous GCD)	`arr[i]`	`gcd(result, arr[i])`
1	12	24	gcd(12, 24) = 12
2	12	36	gcd(12, 36) = 12
3	12	48	gcd(12, 48) = 12

Since the GCD never drops to 1, we never hit the `if (result == 1)` shortcut.

📌 Final Output:

```
GCD of the array elements: 12
```

GCD of the array elements: 12