Array

1}Given three arrays sorted in increasing order. Find the elements that are common in all three arrays.

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

int main() {
    int n1, n2, n3;
    cout << "Enter size of first array: ";
    cin >> n1;
    cout << "Enter size of second array: ";
    cin >> n2;
    cout << "Enter size of third array: ";
    cin >> n3;

    vector<int> A(n1), B(n2), C(n3);
    cout << "Enter elements of first array (sorted): ";
    for (int i = 0; i < n1; i++) cin >> A[i];
    cout << "Enter elements of second array (sorted): ";
    for (int i = 0; i < n2; i++) cin >> B[i];
    cout << "Enter elements of third array (sorted): ";
    for (int i = 0; i < n3; i++) cin >> C[i];

    int i = 0, j = 0, k = 0;
    vector<int> common;

    // Traverse all three arrays together
    while (i < n1 && j < n2 && k < n3) {
```

```
    while (i < n1 && j < n2 && k < n3) {
        if (A[i] == B[j] && B[j] == C[k]) {
            common.push_back(A[i]);
            i++; j++; k++;
        }
        else if (A[i] < B[j])
            i++;
        else if (B[j] < C[k])
            j++;
        else
            k++;
    }

    if (common.empty()) {
        cout << "No common elements found." << endl;
    } else {
        cout << "Common elements: ";
        for (int x : common) cout << x << " ";
        cout << endl;
    }

    return 0;
}
```

Output:

```
Enter size of first array: 2
Enter size of second array: 2
Enter size of third array: 3
Enter elements of first array (sorted): 1
2
Enter elements of second array (sorted): 3
5
Enter elements of third array (sorted): 5
6
7
No common elements found.


=== Code Execution Successful ===
```

2}Given an array of positive and negative numbers. Find if there is a subarray (of size at least one) with 0 sum.

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

int main() {
    int n;
    cout << "Enter size of array: ";
    cin >> n;

    vector<int> arr(n);
    cout << "Enter array elements (can be positive or negative): ";
    for (int i = 0; i < n; i++)
        cin >> arr[i];

    unordered_set<int> prefixSumSet;
    int prefixSum = 0;
    bool zeroSumExists = false;

    for (int i = 0; i < n; i++) {
        prefixSum += arr[i];

        // Check for zero-sum subarray
        if (prefixSum == 0 || prefixSumSet.find(prefixSum) !=
            prefixSumSet.end()) {
            zeroSumExists = true;
```

```cpp
        // Check for zero-sum subarray
        if (prefixSum == 0 || prefixSumSet.find(prefixSum) !=
            prefixSumSet.end()) {
            zeroSumExists = true;
            break;
        }

        prefixSumSet.insert(prefixSum);
    }

    if (zeroSumExists)
        cout << "Yes, there exists a subarray with 0 sum." << endl;
    else
        cout << "No, there is no subarray with 0 sum." << endl;

    return 0;
}
```

OUTPUT:

```
Enter size of array: 2
Enter array elements (can be positive or negative): 34
45
No, there is no subarray with 0 sum.


=== Code Execution Successful ===
```

3}Given an array of size n and an integer k, find all elements in the array that appear more than n/k times.

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

int main() {
    int n, k;
    cout << "Enter size of array: ";
    cin >> n;

    vector<int> arr(n);
    cout << "Enter array elements: ";
    for (int i = 0; i < n; i++)
        cin >> arr[i];

    cout << "Enter value of k: ";
    cin >> k;

    unordered_map<int, int> freq;
    for (int num : arr)
        freq[num]++;

    cout << "Elements that appear more than n/k times are: ";
    bool found = false;
    for (auto it : freq) {
        if (it.second > n / k) {
```

```
22
23      cout << "Elements that appear more than n/k times are: ";
24      bool found = false;
25      for (auto it : freq) {
26          if (it.second > n / k) {
27              cout << it.first << " ";
28              found = true;
29          }
30      }
31
32      if (!found)
33          cout << "None";
34      cout << endl;
35
36      return 0;
37  }
```

OUTPUT:

```
Enter size of array: 3
Enter array elements: 56
67
89
Enter value of k: 4
Elements that appear more than n/k times are: 89 67 56


=== Code Execution Successful ===
```

4}Reverse the given array or string.

```cpp
#include <iostream>
#include <vector>
#include <algorithm> // for reverse()
using namespace std;

int main() {
    int n;
    cout << "Enter size of array: ";
    cin >> n;

    vector<int> arr(n);
    cout << "Enter array elements: ";
    for (int i = 0; i < n; i++)
        cin >> arr[i];

    // Reverse the array
    reverse(arr.begin(), arr.end());

    cout << "Reversed array: ";
    for (int x : arr)
        cout << x << " ";
    cout << endl;

    return 0;
}
```

OUTPUT:

```
Enter size of array: 3
Enter array elements: 45
67
78
Reversed array: 78 67 45


=== Code Execution Successful ===
```