

Objective I. Unique elements

At the input to the program, the value $n < 100$ is first fed - the number of elements in the array.
The next line of the input data contains the elements of the array themselves - integers, modulo not exceeding 30000.
Print only those values of the array elements that occur exactly once in it.
Items should be printed in the order in which they appear in the array.

Examples of
input data

8
4 3 5 2 5 1 3 5

output
4 2 1

```
void checkDoubles() {
    int arr[] = {3, 87, 0, 3, 9, 14, 87, 0, 15};
    int flag;

    for (int i = 0; i < 9; i++) { // loop goes through array w/ each #
        flag = 0;
        for (int j = 0; j < 9; j++) { // loop checks arr[i] # in array again for doubles
            if (arr[i] == arr[j] && i != j) {
                flag = 1; // flag says "this is a double"
                break; // if encountered double, may as well end this iteration of loop
            }
        }
        if (flag == 0) { // if number not flagged, then it had no doubles in array
            cout << arr[i] << " " << endl;
        }
    }
}
```



```
void checkDoubles() {
    int arr[] = {3, 87, 0, 3, 9, 14, 87, 0, 15};
    vector<int> doubles;
    vector<int>::iterator it;
    int flag;

    for (int i = 0; i < 9; i++) { // loop goes through array w/ each #
        flag = 0;
        for (int j = 0; j < 9; j++) { // loop checks arr[i] # in array again for doubles
            if (arr[i] == arr[j] && i != j) {
                flag = 1; // flag says "this is a double"
                it = find (doubles.begin(), doubles.end(), arr[j]);
                if (it == doubles.end()) {
                    doubles.push_back(arr[j]);
                }
                break; // if encountered double, may as well end this iteration of loop
            }
        }
    }
    for (int j = 0; j < doubles.size(); j++) {
        cout << doubles[j] << " ";
    }
    cout << endl;
}
```