

题目描述：

1. 给定一个乱序的数组，删除所有的重复元素，使得每个元素只出现一次，并且按照出现的次数从高到低进行排序，相同出现次数按照第一次出现顺序进行先后排序。

输入描述：

一个数组

输出描述：

去重排序后的数组

补充说明：

数组大小不超过 100

数组元素值大小不超过 100

示例 1

输入：

1, 3, 3, 3, 2, 4, 4, 4, 5

输出：

3, 4, 1, 2, 5

说明：

```
#include <algorithm>
#include <iostream>
#include <iterator>
#include <set>
#include <array>
#include <list>
using namespace std;

void print( std::list<int>& ls) {
    for (auto& it : ls) {
        std::cout << it << " ";
    }
    std::cout << std::endl;
```

```

}

int main() {
    std::array<int, 101> arr = {0};

    int tmp;
    char ctmp;
    std::set<int> had;
    std::list<int> sortls;
    while (cin >> tmp) { // 注意 while 处理多个 case
        arr[tmp]++;
        if (had.emplace(tmp).second)
            sortls.emplace_back(tmp);
        if (cin >> ctmp)
            continue;
    }

    std::list<int> ans;
    ans.assign(had.begin(), had.end());
    // print(ans);
    // for(int i = 0; i < 10; i++)
    // {
    //     std::cout << arr[i] << " ";
    // }
    //std::cout << std::endl;
    ans.sort([&](int& a, int& b) {
        if (arr[a] != arr[b])
            return arr[a] > arr[b];

        auto ita = std::find_if(sortls.begin(), sortls.end(), [&](auto & x) {
            return x == a;
        });
    });
}

```

```

        auto itb = std::find_if(sortls.begin(), sortls.end(), [&](auto & x) {
            return x == b;
        });
        auto da = std::distance(sortls.begin(), ita);
        auto db = std::distance(sortls.begin(), itb);
        return da < db;
    });
    bool is_first(true);
    for (auto& it : ans) {
        if (!is_first)
            std::cout << ", ";
        else
            is_first = false;
        std::cout << it;
    }
    std::cout << endl;
}

// 64 位输出请用 printf("%lld")

```