1.

```
#include <iostream>
#include <algorithm>
#include <iterator>
#include <vector>
using namespace std;
int main() {
    int arr[] = { -2, 19, 80, -47, 80, 80, -2 };
    int* endd = arr + sizeof(arr) / sizeof(arr[0]);
    copy(arr, endd, ostream_iterator<int>(cout, " "));
    /*
    cout << endl;
```

```
PS C:\Users\KanataLi\OneDrive\文档\23 Fall\CSE 332(技术
选修 C++)\Algorithm Studio\Algorithm Studio> g++ -O2 .\
ain.cpp -o main
PS C:\Users\KanataLi\OneDrive\文档\23 Fall\CSE 332(技术
选修 C++)\Algorithm Studio\Algorithm Studio> ./main
-2 19 80 -47 80 80 -2
PS C:\Users\KanataLi\OneDrive\文档\23 Fall\CSE 332(技术
选修 C++)\Algorithm Studio\Algorithm Studio>
```

2.

```
#include <iostream>
#include <algorithm>
#include <iterator>
#include <vector>
using namespace std;
int main() {
    int arr[] = { -2, 19, 80, -47, 80, 80, -2 };
    int* endd = arr + sizeof(arr) / sizeof(arr[0]);
    copy(arr, endd, ostream_iterator<int>(cout, " "));
    cout << endl;
    vector<int> vec;
    back_insert_iterator< vector<int> > backIt(vec);
    copy(arr, endd, backIt);
    copy(vec.begin(), vec.end(), ostream_iterator<int>(cout, " ")
    cout << endl;
    /*
    sort(arr, end(arr));
    sort(vec.begin(), vec.end());
```

```
PS C:\Users\KanataLi\OneDrive\文档\23 Fall\CSE 332(技术
选修 C++)\Algorithm Studio\Algorithm Studio> g++ -O2 .
ain.cpp -o main
PS C:\Users\KanataLi\OneDrive\文档\23 Fall\CSE 332(技术
选修 C++)\Algorithm Studio\Algorithm Studio> ./main
-2 19 80 -47 80 80 -2
-2 19 80 -47 80 80 -2

PS C:\Users\KanataLi\OneDrive\文档\23 Fall\CSE 332(技术
选修 C++)\Algorithm Studio\Algorithm Studio>
```

3.

```
int main() {
    int arr[] = { -2, 19, 80, -47, 80, 80, -2 };
    int* endd = arr + sizeof(arr) / sizeof(arr[0]);
    copy(arr, endd, ostream_iterator<int>(cout, " "));
    cout << endl;
    vector<int> vec;
    back_insert_iterator< vector<int> > backIt(vec);
    copy(arr, endd, backIt);
    copy(vec.begin(), vec.end(), ostream_iterator<int>(cout, " ")
    cout << endl;
    sort(arr, end(arr));
    sort(vec.begin(), vec.end());
    cout << "Sorted array: ";
    copy(arr, arr + sizeof(arr)/sizeof(arr[0]), ostream_iterator<
    cout << endl;
    cout << "Sorted vector: ";
    copy(vec.begin(), vec.end(), ostream_iterator<int>(cout, " ")
    cout << endl;

    return 0;
}
```

```
PS C:\Users\KanataLi\OneDrive\文档\23 Fall\CSE 332(技术
选修 C++)\Algorithm Studio\Algorithm Studio> g++ -O2 .
ain.cpp -o main
PS C:\Users\KanataLi\OneDrive\文档\23 Fall\CSE 332(技术
选修 C++)\Algorithm Studio\Algorithm Studio> ./main
-2 19 80 -47 80 80 -2
-2 19 80 -47 80 80 -2
Sorted array: -47 -2 -2 19 80 80 80
Sorted vector: -47 -2 -2 19 80 80 80
PS C:\Users\KanataLi\OneDrive\文档\23 Fall\CSE 332(技术
选修 C++)\Algorithm Studio\Algorithm Studio>
```

4.

Algorithm Studio (全局范围) findRepeat<Iterato

```cpp
#include <iostream>
#include <algorithm>
#include <iterator>
#include <vector>
#include <set>
using namespace std;
template <typename Iterator>  <T> 提供 IntelliSense 的示例模板参数
inline void findRepeat(Iterator start, Iterator end) {
    //set<typename iterator_traits<Iterator>::value_type> seen;
    while (start != end) {
        //if (seen.count(*start)) {
    //{       ++start;
        //      continue;
    //{ }
        auto repeat = adjacent_find(start, end);
        if (repeat == end) break;
        auto next = find_if(repeat + 1, end, [repeat](const auto& x) { return x != *repeat; });
        cout << *repeat << " ";
        for (auto it = repeat + 1; it != next; ++it) {
            cout << *it << " ";
        }
        cout << endl;
       // seen.insert(*repeat);
        start = next;
    }
}
```

```
选修 C++)\Algorithm Studio\Algorithm Studio> g++ -O2 .\
ain.cpp -o main
PS C:\Users\KanataLi\OneDrive\文档\23 Fall\CSE 332(技术
选修 C++)\Algorithm Studio\Algorithm Studio> ./main
-2 19 80 -47 80 80 -2
-2 19 80 -47 80 80 -2
Sorted array: -47 -2 -2 19 80 80 80
Sorted vector: -47 -2 -2 19 80 80 80
-2 -2
80 80 80
-2 -2
80 80 80
PS C:\Users\KanataLi\OneDrive\文档\23 Fall\CSE 332(技术
选修 C++)\Algorithm Studio\Algorithm Studio>
```

igorithm Studio (全局范围)

```cpp
    copy(arr, arr + sizeof(arr)/sizeof(arr[0]), ostream_iterator<int>
    cout << endl;
    cout << "Sorted vector: ";
    copy(vec.begin(), vec.end(), ostream_iterator<int>(cout, " "));
    cout << endl;
    findRepeat(begin(arr), end(arr));
    findRepeat(vec.begin(), vec.end());
    return 0;
}
```


```
PS C:\Users\KanataLi\OneDrive\文档\23 Fall\CSE 332(技术
选修 C++)\Algorithm Studio\Algorithm Studio> g++ -O2 .\m
ain.cpp -o main
PS C:\Users\KanataLi\OneDrive\文档\23 Fall\CSE 332(技术
选修 C++)\Algorithm Studio\Algorithm Studio> ./main
-2 19 80 -47 80 80 -2
-2 19 80 -47 80 80 -2
Sorted array: -47 -2 -2 19 80 80 80
Sorted vector: -47 -2 -2 19 80 80 80
-2 -2
80 80 80
-2 -2
80 80 80
PS C:\Users\KanataLi\OneDrive\文档\23 Fall\CSE 332(技术
选修 C++)\Algorithm Studio\Algorithm Studio>
```

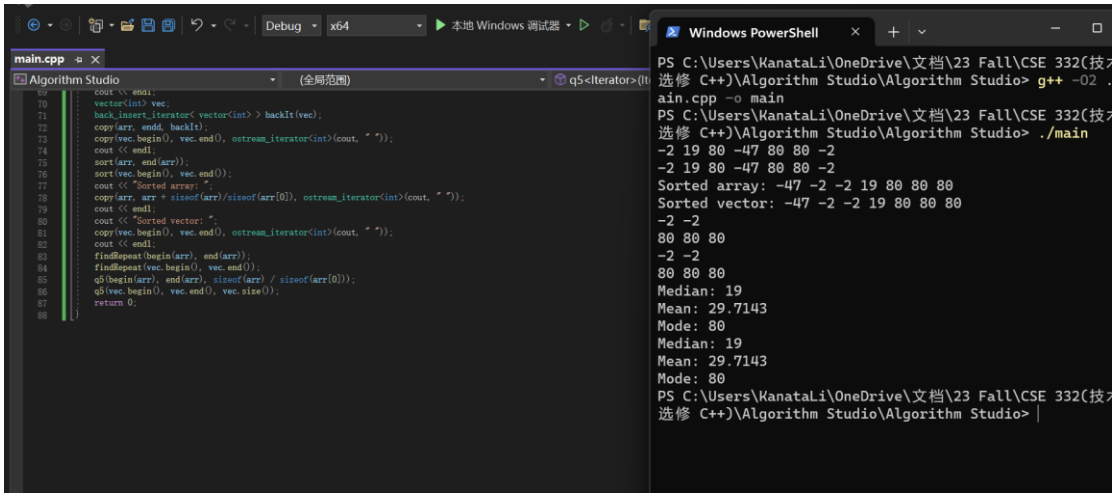

5.

```cpp
template <typename Iterator>  <T> 提供 IntelliSense 的示例模板参数
inline void q5(Iterator start, Iterator end, int size) {
    cout << "Median: ";
    if (size & 1) {
        cout << *(start + (size >> 1)) << endl; // Odd size
    }
    else {
        cout << (*(start + (size / 2 - 1)) + *(start + (size / 2))) / 2.0 << endl; // Even size
    }

    float mean = accumulate(start, end, 0.0f) / size;
    cout << "Mean: " << mean << endl;

    vector<pair<int, typename iterator_traits<Iterator>::value_type>> v;
    set<typename iterator_traits<Iterator>::value_type> seen;

    for (Iterator it = start; it != end; ++it) {
        if (seen.count(*it) == 0) {
            v.push_back({ count(start, end, *it), *it });
            seen.insert(*it);
        }
    }
    sort(v.begin(), v.end(), [](const auto& a, const auto& b) { return a.first > b.first; });
    cout << "Mode: ";
    if (!v.empty()) {
        int maxFrequency = v.front().first;
        for (const auto& pair : v) {
            if (pair.first == maxFrequency) {
                cout << pair.second << " ";
            }
            else {
                break;
            }
        }
    }
    cout << endl;
}
int main() {
```


Debug
x64
本地 Windows 调试器
main.cpp
Algorithm Studio
(全局范围)
cout << endl;
vector<int> vec;
back_insert_iterator< vector<int> > backIt(vec);
copy(arr, endd, backIt);
copy(vec.begin(), vec.end(), ostream_iterator<int>(cout, " "));
cout << endl;
sort(arr, end(arr));
sort(vec.begin(), vec.end());
cout << "Sorted array: ";
copy(arr, arr + sizeof(arr)/sizeof(arr[0]), ostream_iterator<int>(cout, " "));
cout << endl;
cout << "Sorted vector: ";
copy(vec.begin(), vec.end(), ostream_iterator<int>(cout, " "));
cout << endl;
findRepeat(begin(arr), end(arr));
findRepeat(vec.begin(), vec.end());
q5(begin(arr), end(arr), sizeof(arr) / sizeof(arr[0]));
q5(vec.begin(), vec.end(), vec.size());
return 0;
}
Windows PowerShell
PS C:\Users\KanataLi\OneDrive\文档\23 Fall\CSE 332(技术选修 C++)\Algorithm Studio\Algorithm Studio> g++ -O2 .
ain.cpp -o main
PS C:\Users\KanataLi\OneDrive\文档\23 Fall\CSE 332(技术选修 C++)\Algorithm Studio\Algorithm Studio> ./main
-2 19 80 -47 80 80 -2
-2 19 80 -47 80 80 -2
Sorted array: -47 -2 -2 19 80 80 80
Sorted vector: -47 -2 -2 19 80 80 80
-2 -2
80 80 80
-2 -2
80 80 80
Median: 19
Mean: 29.7143
Mode: 80
Median: 19
Mean: 29.7143
Mode: 80
PS C:\Users\KanataLi\OneDrive\文档\23 Fall\CSE 332(技术选修 C++)\Algorithm Studio\Algorithm Studio>


不安全
8.130.92.7/success/class/homework/problem
5 of Clubs
5 of Dia
5 of Hea
5 of Spa
6 of Clu
6 of Dia
6 of Hea
6 of Spa
7 of Clu
7 of Dia
7 of Hea
7 of Spa
8 of Clu
8 of Dia
8 of Hea
8 of Spa
9 of Clu
9 of Dia
9 of Hea
9 of Spa
10 of Cl
10 of Di
10 of He
10 of Sp
Jack of
编号
运行时间
占用内存
得分
操作
1
0m0.010s
6784kb
100
请点这里查看历史记录或报错
点这里关闭本窗口

7.

```
bool checkAceLowStraight(const vector<int>& sortedRanks) {
    return sortedRanks[0] == static_cast<int>(Rank::TWO) &&
           sortedRanks[1] == static_cast<int>(Rank::THREE) &&
           sortedRanks[2] == static_cast<int>(Rank::FOUR) &&
           sortedRanks[3] == static_cast<int>(Rank::FIVE) &&
           sortedRanks.back() == static_cast<int>(Rank::ACE);
}
vector<Card> drawHand(vector<Card>& deck) {
    shuffle(deck.begin(), deck.end(), default_random_engine(random_device{}()));
    vector<Card> hand(deck.end() - 5, deck.end());
    deck.erase(deck.end() - 5, deck.end());
    return hand;
}
string evaluateHand(const vector<Card>& hand) {
    map<Rank, int> rankFrequency;
    map<Suit, int> suitFrequency;
    for (const auto& card : hand) {
        rankFrequency[card.rank]++;
        suitFrequency[card.suit]++;
    }

    // Check for flush, straight, etc.
    bool isFlush = any_of(suitFrequency.begin(), suitFrequency.end(),
                          [&](const auto& p) { return p.second == 5; });

    vector<int> sortedRanks;
    for (const auto& p : rankFrequency) {
        sortedRanks.push_back(static_cast<int>(p.first));
    }
    sort(sortedRanks.begin(), sortedRanks.end());

    bool isStraight = true;
    for (size_t i = 1; i < sortedRanks.size() && isStraight; ++i) {
        if (sortedRanks[i] != sortedRanks[i - 1] + 1) {
            isStraight = false;
        }
    }
    if(!isStraight) isStraight = checkAceLowStraight(sortedRanks);
    int pairs = 0, threeOfAKind = 0, fourOfAKind = 0;
    for (const auto& p : rankFrequency) {
        if (p.second == 2) pairs++;
        if (p.second == 3) threeOfAKind++;
        if (p.second == 4) fourOfAKind++;
    }

    if (isStraight && isFlush) return "Straight Flush";
    if (fourOfAKind) return "Four of a Kind";
    if (threeOfAKind && pairs) return "Full House";
    if (isFlush) return "Flush";
    if (isStraight) return "Straight";
    if (threeOfAKind) return "Three of a Kind";
    if (pairs == 2) return "Two Pair";
    if (pairs) return "One Pair";
    return "High Card";
}
```

```
    // with no replacement
    for(int i = 1; i <= 10; ++i) {
        cout << "Pair " << i << endl;
        vector<Card> now = drawHand(deck);
        sort(now.begin(), now.end());
        for(const auto& card : now) {
            cout << card << endl;
        }
        cout << "is " << evaluateHand(now) << endl;
    }
```

```
PS C:\Users\KanataLi\OneDrive\文档\23 Fall\CSE 332(技术选修 C++)\Algorithm Studio\Algorithm Studio\enh> g++ -O2 .\main.cpp -o main
PS C:\Users\KanataLi\OneDrive\文档\23 Fall\CSE 332(技术选修 C++)\Algorithm Studio\Algorithm Studio\enh>
PS C:\Users\KanataLi\OneDrive\文档\23 Fall\CSE 332(技术选修 C++)\Algorithm Studio\Algorithm Studio\enh> .\main
Pair 1
3 of Clubs
4 of Hearts
8 of Spades
9 of Diamonds
Queen of Diamonds
is High Card
Pair 2
3 of Hearts
5 of Spades
10 of Clubs
Queen of Hearts
Queen of Spades
is One Pair
Pair 3
4 of Spades
5 of Clubs
6 of Hearts
8 of Clubs
Ace of Hearts
is High Card
Pair 4
4 of Diamonds
6 of Diamonds
10 of Diamonds
Jack of Diamonds
Queen of Clubs
is High Card
Pair 5
2 of Hearts
Jack of Spades
King of Diamonds
Ace of Clubs
Ace of Spades
is One Pair
Pair 6
2 of Clubs
7 of Spades
Jack of Hearts
King of Hearts
Ace of Diamonds
is High Card
Pair 7
2 of Diamonds
4 of Clubs
8 of Diamonds
8 of Hearts
10 of Hearts
is One Pair
Pair 8
2 of Spades
3 of Spades
9 of Hearts
10 of Spades
King of Clubs
is High Card
Pair 9
5 of Hearts
6 of Clubs
7 of Clubs
9 of Spades
Jack of Clubs
is High Card
Pair 10
3 of Diamonds
5 of Diamonds
6 of Spades
7 of Diamonds
7 of Hearts
is One Pair
PS C:\Users\KanataLi\OneDrive\文档\23 Fall\CSE 332(技术选修 C++)\Algorithm Studio\Algorithm Studio\enh>
```