

1. 任意输入 N 个正整数，找到其中出现频率最高的数，以及他的次数，如果有两个相同频率的数，则去最小的那个数 10 分

```
1 #include <iostream>
2 #include <unordered_map>
3 using namespace std;
4
5 int main() {
6     int N;
7     cout << "请输入正整数的个数： ";
8     cin >> N;
9
10    // 使用 unordered_map 来记录每个数字的出现次数
11    unordered_map<int, int> frequency;
12
13    // 用户输入 N 个数，记录数字次数到 frequency 中
14    cout << "请输入 " << N << " 个正整数： ";
15    for (int i = 0; i < N; i++) {
16        int num;
17        cin >> num;
18        frequency[num]++;
19    }
20
21    // 找出出现次数最多的数字以及次数
22    int maxCount = 0;
23    int mostFrequentNum = 0;
24
25    for (const auto& pair : frequency) {
26        if (pair.second > maxCount || (pair.second == maxCount && pair.first < mostFrequentNum))
27        {
28            maxCount = pair.second;
29            mostFrequentNum = pair.first;
30        }
31    }
32
33    // 输出数字及出现的次数
34    cout << mostFrequentNum << " :: " << maxCount << endl;
35
36    return 0;
37 }
```

2. 任意输入 N 个不为 0 的数，求其相反数的对数。

```
1 #include <iostream>
2 #define N 6
3 using namespace std;
4
5 int main() {
6     int a[N];
7     int count = 0, i, j;
8
9     // 用户输入数字
10    for (i = 0; i < N; i++) {
11        cin >> a[i];
12    }
13
14    // 开始从第一个向后找是否是相反数的数字
15    for (i = 0; i < N - 1; i++) {
16        // 每次从其下一个开始找
17        for (j = i + 1; j < N; j++) {
18            if (a[i] == -a[j]) {
```

```

19         count++;
20         break;
21     }
22 }
23 }
24
25 // 输出相反数的对数
26 cout << count << endl;
27 return 0;
28 }

```

3. ftp 原题改为 ftp://ouc:test@116.119.0.1:2121

4. 矩阵题 自定义输入输出，重载运算符 以及实现 43 矩阵和 3\*4 矩阵的乘法（要求矩阵各数随机输入）

```

1  #include <iostream>
2  using namespace std;
3
4  class Juzhen {
5  public:
6      // 构造函数初始化矩阵行列
7      Juzhen(const int R, const int C) : row(R), col(C) {
8          int i, j;
9          a = new int*[row];
10         for (i = 0; i < row; i++) {
11             a[i] = new int[col];
12         }
13         for (i = 0; i < row; i++) {
14             for (j = 0; j < col; j++) {
15                 a[i][j] = 0;
16             }
17         }
18     }
19
20     // 拷贝构造函数
21     Juzhen(const Juzhen& j1) {
22         row = j1.row;
23         col = j1.col;
24         int i, j;
25         a = new int*[row];
26         for (i = 0; i < row; i++) {
27             a[i] = new int[col];
28         }
29         for (i = 0; i < row; i++) {
30             for (j = 0; j < col; j++) {
31                 a[i][j] = j1.a[i][j];
32             }
33         }
34     }
35
36     // 析构函数释放内存空间
37     ~Juzhen() {
38         for (int i = 0; i < row; i++) {
39             delete[] a[i];
40         }
41         delete[] a;
42     }
43
44     // 随机生成矩阵值
45     void input();
46
47     // 显示当前矩阵值

```

```
48     void show() const;
49
50     // 重载=操作符
51     Juzhen& operator=(const Juzhen&);
52
53     // 重载*操作符
54     Juzhen operator*(const Juzhen&);
55
56 private:
57     int** a;
58     int row, col;
59 };
60
61 Juzhen Juzhen::operator*(const Juzhen& j1) {
62     Juzhen j2(row, j1.col);
63     if (col == j1.row) {
64         for (int i = 0; i < row; i++) {
65             for (int j = 0; j < j1.col; j++) {
66                 int sum = 0;
67                 for (int k = 0; k < col; k++) {
68                     sum += a[i][k] * j1.a[k][j];
69                 }
70                 j2.a[i][j] = sum;
71             }
72         }
73     }
74     return j2;
75 }
76
77 Juzhen& Juzhen::operator=(const Juzhen& j1) {
78     if (this == &j1) {
79         return *this;
80     }
81     int i, j;
82     for (i = 0; i < row; i++) {
83         delete[] a[i];
84     }
85     delete[] a;
86     row = j1.row;
87     col = j1.col;
88     a = new int*[row];
89     for (i = 0; i < row; i++) {
90         a[i] = new int[col];
91     }
92     for (i = 0; i < row; i++) {
93         for (j = 0; j < col; j++) {
94             a[i][j] = j1.a[i][j];
95         }
96     }
97     return *this;
98 }
99
100 void Juzhen::show() const {
101     cout << "当前矩阵为: " << endl;
102     for (int i = 0; i < row; i++) {
103         for (int j = 0; j < col; j++) {
104             cout << a[i][j] << " ";
105         }
106         cout << endl;
107     }
108 }
```

```
109
110 void Juzhen::input() {
111     cout << "输入矩阵值: " << endl;
112     for (int i = 0; i < row; i++) {
113         for (int j = 0; j < col; j++) {
114             cin >> a[i][j];
115         }
116     }
117 }
118
119 int main() {
120     // 分别创建矩阵 A,B
121     Juzhen A(3, 4);
122     A.input();
123     Juzhen B(4, 3);
124     B.input();
125     Juzhen C(3,3);
126     //矩阵相乘
127     C=A*B;
128     C.show();
129     return 0;
130 }
```

5. 15 年微信原题