set和map

人员

辛帅辰、方俊喆、刘祺、夏硕承、秦显森、齐振玮、徐浩然、谢亚锴、牛同泽、牟茗、高健桓、赵熙羽 到课

作业检查

作业链接: https://www.luogu.com.cn/contest/198272 (E F 2道题)

韩承睿 完成 2 道

赵熙羽、辛帅辰 完成 1 道

其他同学未做

作业

https://www.luogu.com.cn/contest/199446 (E F 2道题目)

课堂表现

同学们课上听课都很认真，课下要好好复习 set 和 map 的使用, 以及上节课的 queue 和 stack

课堂内容

U476049 数组填充

```
#include <bits/stdc++.h>
using namespace std;

const int maxn = 500 + 5;
int a[maxn][maxn];

int main() {
    int n; cin >> n;
    int id = 1;
    for (int k = 2*n; k >= 2; k--) {
        if (k%2 == 1) {
            for (int i = 1; i <= n; i++) {
                for (int j = 1; j <= n; j++) {
                    if (i+j == k) {
                        a[i][j] = id;
                        id++;
                    }
                }
            }
        }
        else {
            for (int i = n; i >= 1; i--) {
                for (int j = 1; j <= n; j++) {
                    if (i+j == k) {
                        a[i][j] = id;
                        id++;
                    }
                }
            }
        }
    }

    for (int i = 1; i <= n; i++) {
        for (int j = 1; j <= n; j++) {
            cout << a[i][j] << " ";
        }
        cout << endl;
    }
    return 0;
}
```

U478303 有多少个可能的密码数量

```
#include <bits/stdc++.h>
using namespace std;

bool check(int a, int b, int c, int d, string s) {
    int f[10];
    memset(f, 0, sizeof(f));
    f[a]++, f[b]++, f[c]++, f[d]++;

    for (int i = 0; i < 10; i++) {
        if (s[i] == 'o') {
            if (f[i] == 0) return false;
        }
        if (s[i] == 'x') {
            if (f[i] != 0) return false;
        }
    }
    return true;
}

int main()
{
    string s; cin >> s;
    int cnt = 0;
    for (int i = 0; i <= 9999; i++) {
        int a = i/1000, b = (i/100)%10, c = (i/10)%10, d = i%10;
        if (check(a, b, c, d, s)) {
            cnt++;
        }
    }
    cout << cnt << endl;
    return 0;
}
```

P1059 [NOIP2006 普及组] 明明的随机数

```
#include <bits/stdc++.h>
using namespace std;

const int maxn = 1000 + 5;
int f[maxn];  // f[i]: i 出现的次数

int main() {
    int n; cin >> n;
    while (n -- ) {
        int x; cin >> x; f[x]++;
    }

    int cnt = 0;
    for (int i = 1; i <= 1000; i++) {
        if (f[i] != 0) {
            cnt++;
        }
    }
    cout << cnt << endl;

    for (int i = 1; i <= 1000; i++) {
        if (f[i] != 0) {
            cout << i << " ";
        }
    }
    return 0;
}
```

U480360 明明的随机数2

```
// 方法一
#include <bits/stdc++.h>
using namespace std;

const int maxn = 1e5 + 5;
int w[maxn];

int main() {
    int n; cin >> n;
    for (int i = 1; i <= n; ++i) cin >> w[i];
    sort(w+1, w+n+1);

    int cnt = 0;
    for (int i = 1; i <= n; i++) {
        if (w[i] != w[i-1]) {
            cnt++;
        }
    }
    cout << cnt << endl;

    for (int i = 1; i <= n; i++) {
        if (w[i] != w[i-1]) {
            cout << w[i] << " ";
        }
    }
    return 0;
}
```

```
// 方法二
#include <bits/stdc++.h>

using namespace std;

int main()
{
    set<int>s;
    int n; cin >> n;
    while (n -- ) {
        int x; cin >> x; s.insert(x);
    }
    cout << s.size() << endl;
    for (int i : s) cout << i << " ";
    return 0;
}
```

set

```
#include <set>
set<int> s;
s.insert(7);  // 插入
s.erase(7);  // 删除
s.empty();  // 判断是否为空
s.size();  // 求 set 的大小
s.count(7);  // 判断 set 里面是否有 7
s.clear();  // set 清空
for (int i : s) {  // 遍历 set, 需要开 c++11
    cout << i << " ";
}
```

U480367 A-B数对(简易版)

```cpp
#include <bits/stdc++.h>
using namespace std;

const int N = 2e5+5, M = 1e6+5;
int w[N], f[M];

int main() {
    int n, C; cin >> n >> C;
    for (int i = 1; i <= n; ++i) {
        cin >> w[i];
        f[w[i]]++;
    }

    long long res = 0;
    for (int i = 1; i <= n; i++) {
        int A = w[i];
        int B = A - C;
        if (B >= 0) res += f[B];
    }

    cout << res << endl;
    return 0;
}
```

P1102 A-B 数对

```cpp
#include <bits/stdc++.h>
using namespace std;

const int N = 2e5+5, M = 1e6+5;
int w[N];
map<int,int> f;

int main() {
    int n, C; cin >> n >> C;
    for (int i = 1; i <= n; ++i) {
        cin >> w[i];
        f[w[i]]++;
    }

    long long res = 0;
    for (int i = 1; i <= n; i++) {
        int A = w[i];
        int B = A - C;
        if (B >= 0) res += f[B];
    }

    cout << res << endl;
    return 0;
}
```

map使用

```
#include <map>
map<int, int> mp;  // mp 可以按照第一关键字排序
mp[3]++;
mp[-20] = 7;
mp[100000000] = 3;
mp.size();  // 求 mp 的大小
for (auto it : mp) {  // 遍历 mp 的所有元素
    cout << it.first << " " << it.second << endl;
}
mp.erase(3);  // 删除 3 对应的信息
if (mp[-20] == 0) {  // 判断 mp 中有没有 -20 这个元素的信息

}
```