

# 综合练习

## 人员

柳力玮、李瑞涵、刘宸熙、初锦阳、韩昱辰、温郝冬、田心一、高健桓、李知朔、蒋叔璋、宋云朗 到课

## 上周作业检查

上周作业链接: <https://cppoj.kids123code.com/contest/1117>

### 王向东老师周日三点半C++综合练习

#	用户名	姓名	编程分	时间	A	B	C	D
1	hanyuchen	韩昱辰	400	11575	100	100	100	100
2	liuliwei	柳力玮	400	14161	100	100	100	100
3	lizishuo	李知朔	320	15004	100	100	100	20
4	gaojianhuan	高健桓	313	10793	100	100	13	100
5	yuanzhao	苑钊	300	24808	100	100	100	
6	tianxinyi	田心一	300	25351	100	100	100	
7	chujin yang	初锦阳	204	17374	100	100	4	
8	jiangshuzhang	蒋叔璋	200	3336	100	100		
9	liruihan	李瑞涵	200	3576	100	100		
10	zhaomuzhi	赵牧之	200	10722	100	100		
11	liuchenxi	刘宸熙	200	12192	100	100		
12	wenhaodong	温郝冬	100	10607	100	0		
13	wangxinqi	王馨琪	100	10668	100			

## 本周作业

<https://cppoj.kids123code.com/contest/1211> (课上讲了 A ~ D 题, 课后作业是 D 题)

## 课堂表现

今天的题整体来说不是很难, 给同学们复习了 重载运算符、差分、二维前缀和 等几个知识点, 这几个知识点同学们课下要多复习一下。

## 课堂内容

### Chocolate Eating S (上周作业)

二分答案即可, 最后确定答案之后记得再调用一次 check

```
#include <bits/stdc++.h>

using namespace std;

typedef long long LL;
const int maxn = 5e4 + 5;
int w[maxn], f[maxn];
```

```

int n, D;

bool check(LL mid) {
    LL sum = 0, cnt = 0;
    for (int i = 1; i <= n; ++i) {
        sum += w[i]; f[i] = cnt+1;
        while (sum >= mid) ++cnt, sum /= 2;
    }
    return cnt >= D;
}

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

    LL l = 0, r = 5e10+10;
    while (l <= r) {
        LL mid = (l + r) / 2;
        if (check(mid)) l = mid+1;
        else r = mid-1;
    }
    cout << r << endl;

    check(r);
    for (int i = 1; i <= n; ++i) cout << min(f[i],D) << endl;
    return 0;
}

```

## Election Quick Report

set 里存放每个候选人编号以及候选人的得票数

每次给一个人加票时, 从 set 里删掉这个人的信息, 并重新插入新的信息即可

```

#include <bits/stdc++.h>

using namespace std;

const int maxn = 200000 + 5;
int f[maxn];
struct node {
    int id, cnt;
    bool operator < (const node& p) const {
        if (cnt != p.cnt) return cnt > p.cnt;
        return id < p.id;
    }
};

int main()
{

```

```

set<node> s;
int n, m; cin >> n >> m;
for (int i = 1; i <= n; ++i) s.insert({i, 0});
while (m -- ) {
    int x; cin >> x;
    int t = f[x]; ++f[x];
    auto it = s.find({x, t});
    s.erase(it); s.insert({x, t+1});
    cout << (*s.begin()).id << endl;
}
return 0;
}

```

## [蓝桥杯 2024 省 C] 商品库存管理

先用 差分 + 前缀和，计算每个位置的值

最后的查询，其实就是查 全局0的数量 + l~r中1的数量

```

#include <bits/stdc++.h>

using namespace std;

const int maxn = 3e5 + 5;
int w[maxn], p[maxn], f[maxn];
int l[maxn], r[maxn];

int main()
{
    int n, m; cin >> n >> m;
    for (int i = 1; i <= m; ++i) {
        cin >> l[i] >> r[i]; w[l[i]]++, w[r[i]+1]--;
    }
    for (int i = 1; i <= n; ++i) p[i] = p[i-1] + w[i];

    int res = 0;
    for (int i = 1; i <= n; ++i) {
        if (p[i] == 0) ++res;
        f[i] = f[i-1] + (p[i]==1);
    }

    for (int i = 1; i <= m; ++i) {
        int nums = f[r[i]] - f[l[i]-1];
        cout << res+nums << endl;
    }
    return 0;
}

```

## 平铺图案

二维前缀和直接维护即可

```
#include <bits/stdc++.h>
#define int long long

using namespace std;

const int maxn = 1000 + 5;
char s[maxn][maxn];
int p[maxn][maxn];

int calc(int n, int x, int y) {
    int c1 = x / n, c2 = y / n;
    int sx = x % n, sy = y % n;
    int res = c1 * c2 * p[n][n] + c1 * p[n][sy] + c2 * p[sx][n] + p[sx][sy];
    return res;
}

signed main()
{
    int n, m; cin >> n >> m;
    for (int i = 1; i <= n; ++i) cin >> (s[i]+1);
    for (int i = 1; i <= n; ++i) {
        for (int j = 1; j <= n; ++j) {
            p[i][j] = p[i-1][j] + p[i][j-1] - p[i-1][j-1] + (s[i][j]=='B');
        }
    }

    while (m -- ) {
        int a, b, c, d; cin >> a >> b >> c >> d;
        ++a, ++b, ++c, ++d;
        // cout << "----- ";
        cout << calc(n,c,d) - calc(n,a-1,d) - calc(n,c,b-1) + calc(n,a-1,b-1) << endl;
    }
    return 0;
}
```

## New Place

求最长的后缀长度 len, 要求 s 中后 len 个字符的子串是 t 的子序列

求这个的过程可以从后往前扫, 用双指针实现

```
#include <bits/stdc++.h>

using namespace std;

int main()
{
    int n; cin >> n;
```

```
string s, t; cin >> s >> t;

string s2 = s, t2 = t;
sort(s2.begin(), s2.end()), sort(t2.begin(), t2.end());
if (s2 != t2) { cout << -1 << endl; return 0; }

int cnt = 0;
for (int i = n-1, j = n-1; i >= 0; --i) {
    if (t[i] == s[j]) --j, ++cnt;
}
cout << n - cnt << endl;
return 0;
}
```