

# 综合练习

## 人员

崔宸赫、于家瑞、郭栩睿、陶汇笙、于霄龙、王恩泽 到课

## 上周作业检查

上周作业链接: <https://cppoj.kids123code.com/contest/1107>

### 王向东老师周六十点半C++三分

刷新

#	用户名	姓名	编程分	时间	A	B	C	D
1	taohuisheng	陶汇笙	300	1321	100	100	100	
2	hongchendong	洪晨栋	200	529	100	100		
3	guoxurui	郭栩睿	200	991	100	100		
4	hongchenqi	洪晨祺	200	1122	100	100		
5	yujiarui	于家瑞	200	1170	100	100		
6	cuichenhe	崔宸赫	100	316	100	0		

## 本周作业

<https://cppoj.kids123code.com/contest/1193> (课上讲了 A ~ D 这些题, 课后作业是 D 题)

## 课堂表现

今天上课同学们的整体纪律不是太好, 老师管纪律管了好几次, 希望同学们以后改正。

## 课堂内容

### [CERC1998] 书的复制 (上周作业)

二分, 希望前面的人少抄写, 就是后面的人多抄写, 可以从后往前贪心, 让后面的人多抄写即可

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

using namespace std;

const int maxn = 500 + 5;
int w[maxn], ansl[maxn], ansr[maxn];
int m, k;

bool check(int mid) {
    int sum = 0, cnt = 1;
    ansr[cnt] = m;
    for (int i = m; i >= 1; --i) {
        if (sum + w[i] <= mid) sum += w[i];
        else ansl[cnt] = i+1, ++cnt, sum = w[i], ansr[cnt] = i;
    }
}
```

```

    }
    ans1[cnt] = 1;
    return cnt <= k;
}

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

    int l = 1, r = 1e9;
    while (l <= r) {
        int mid = (l + r) / 2;
        if (check(mid)) r = mid - 1;
        else l = mid + 1;
    }

    check(1);
    for (int i = k; i >= 1; --i) cout << ans1[i] << " " << ansr[i] << endl;
    return 0;
}

```

### [蓝桥杯 2017 省 B] k 倍区间

看区间和是不是  $k$  的倍数, 就是看 2 个前缀和模  $k$  的值是否相同

可以把所有前缀和模  $k$ , 求有多少对相同

```

#include <bits/stdc++.h>

using namespace std;

typedef long long LL;
const int maxn = 1e5 + 5;
int w[maxn];
LL p[maxn];
int f[maxn];

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

    for (int i = 0; i <= n; ++i) f[p[i] % k]++;

    LL res = 0;
    for (int i = 0; i < k; ++i) res += (LL)f[i] * (f[i] - 1) / 2;
    cout << res << 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;
}
```

## Take ABC

类似于括号匹配, 用栈存未删除的字符, 当这个字符是C, 且前两个是B和A的时候, 删除; 否则, 把字符往栈里放

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

using namespace std;

const int maxn = 2e5 + 5;
char s[maxn];

int main()
{
    cin >> (s+1);
    int n = strlen(s+1);
```

```

vector<char> vec; int len = 0;
for (int i = 1; i <= n; ++i) {
    if (s[i]=='C' && len>=2 && vec[len-1]=='B' && vec[len-2]=='A') {
        vec.pop_back(), vec.pop_back(), len -= 2;
    } else vec.push_back(s[i]), ++len;
}

for (char i : vec) cout << i;
cout << endl;
return 0;
}

```

## Ghost Ants

把蚂蚁分为 向左走 和 向右走 的 2 类

针对每个向右的蚂蚁, 看它能碰到多少个向左的蚂蚁, 这个过程可以二分做

```

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

using namespace std;

const int maxn = 2e5 + 5;
char s[maxn];
int w[maxn];
vector<int> vec1, vec2;

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

    for (int i = 1; i <= n; ++i) {
        if (s[i] == '1') vec1.push_back(w[i]);
        else vec2.push_back(w[i]);
    }

    sort(vec2.begin(), vec2.end());

    int res = 0;
    for (int i : vec1) {
        // >=i+1, && <=i+2*T
        int l = lower_bound(vec2.begin(), vec2.end(), i+1) - vec2.begin();
        int r = upper_bound(vec2.begin(), vec2.end(), i+2*T) - vec2.begin() - 1;
        res += r-l+1;
    }
    cout << res << endl;
}

```

```
    return 0;  
}
```