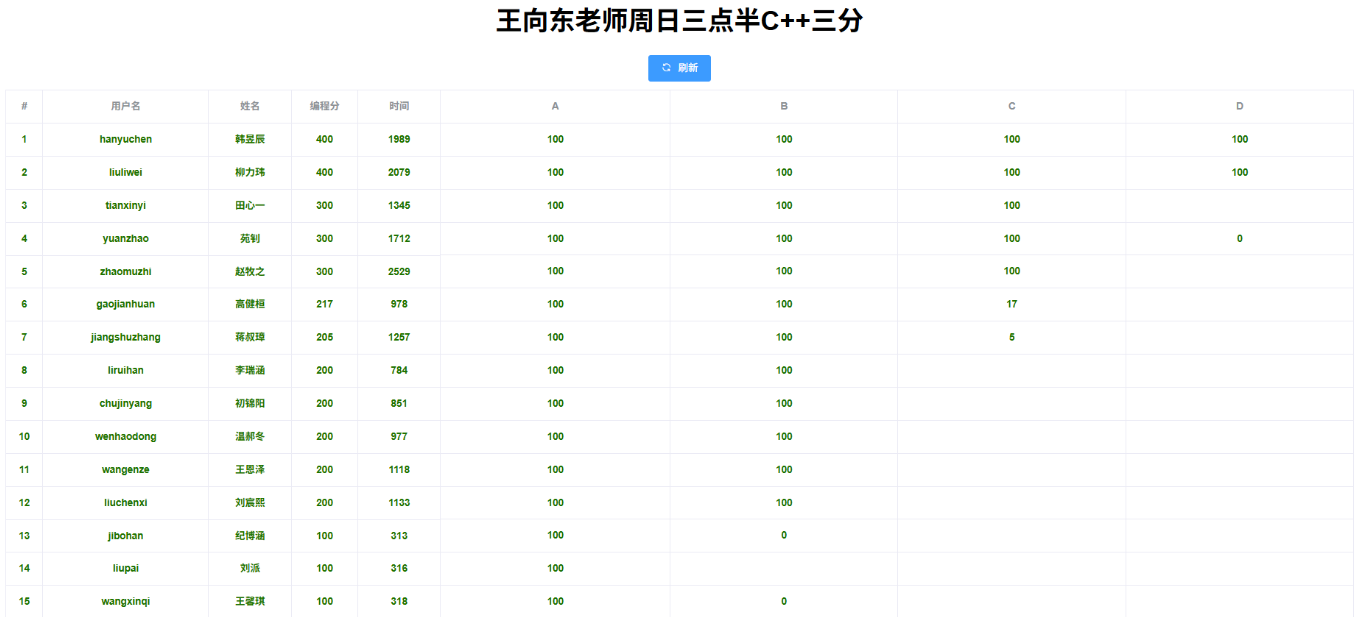
综合练习

人员

柳力玮、李瑞涵、刘宸熙、王馨琪、初锦阳、韩昱辰、苑钊、温郝冬、田心一、赵牧之、高健桓、李知朔 到课, 蒋叔璋 线上

上周作业检查

上周作业链接: https://cppoj.kids123code.com/contest/1031



王向东老师周日三点半C++三分

#	用户名	姓名	编程分	时间	A	B	C	D
1	hanyuchen	韩昱辰	400	1989	100	100	100	100
2	liuliwei	柳力玮	400	2079	100	100	100	100
3	tianxinyi	田心一	300	1345	100	100	100	
4	yuanzhao	苑钊	300	1712	100	100	100	0
5	zhaomuzhi	赵牧之	300	2529	100	100	100	
6	gaojianhuan	高健桓	217	978	100	100	17	
7	jiangshuzhang	蒋叔璋	205	1257	100	100	5	
8	liruihan	李瑞涵	200	784	100	100		
9	chujinyang	初锦阳	200	851	100	100		
10	wenhaodong	温郝冬	200	977	100	100		
11	wangenze	王恩泽	200	1118	100	100		
12	liuchenxi	刘宸熙	200	1133	100	100		
13	jibohan	纪博涵	100	313	100	0		
14	liupai	刘派	100	316	100			
15	wangxinqi	王馨琪	100	318	100	0		

本周作业

https://cppoj.kids123code.com/contest/1117 (课上讲了 A ~ C 题, 课后作业是 D 题)

课堂表现

今天的 A 题比较简单, 同学们一上课做这个题整体都做的不太好, 说明前缀和、二分掌握的不太扎实, 课下要好好复习复习这两个内容

C 题会比较复杂一些, 同学们课下得认真分析、认真做一下这道题。

课堂内容

书的复制 (上周作业)

二分答案, 要求让前面的人少抄写, 所以从后往前进行遍历, 让后面的人抄的越多越好

```
#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;
  }
  ansl[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(l);
  for (int i = k; i >= 1; --i) cout << ansl[i] << " " << ansr[i] << endl;
  return 0;
}
```

1D 国家

直接二分查找, 套前缀和维护查区间和即可

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

using namespace std;

typedef long long LL;
const int maxn = 2e5 + 5;
int X[maxn], P[maxn];
LL pre[maxn];

LL get_sum(int l, int r) { return (l<=r ? pre[r]-pre[l-1] : 0); }

int main()
{
  int n; cin >> n;
  for (int i = 1; i <= n; ++i) cin >> X[i];
  for (int i = 1; i <= n; ++i) cin >> P[i], pre[i] = pre[i-1] + P[i];

  int T; cin >> T;
  while (T -- ) {
    int l, r; cin >> l >> r;
    int pos1 = lower_bound(X+1, X+n+1, l) - X;
    int pos2 = upper_bound(X+1, X+n+1, r) - X - 1;
    cout << get_sum(pos1, pos2) << endl;
  }
  return 0;
}
```

[蓝桥杯 2014 省 AB] 蚂蚁感冒

碰撞反弹等价于穿越过去即可

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

using namespace std;

const int maxn = 50 + 5;
int w[maxn];

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

  int cnt = 0;
    if (w[1] > 0) {
    for (int i = 2; i <= n; ++i) {
      if (w[i]<0 && -w[i]>w[1]) ++cnt;
    }

    if (cnt) {
      for (int i = 2; i <= n; ++i) {
        if (w[i]>0 && w[i]<w[1]) ++cnt;
      }
    }
    } else {
    for (int i = 2; i <= n; ++i) {
      if (w[i]>0 && w[i]<-w[1]) ++cnt;
    }

    if (cnt) {
      for (int i = 2; i <= n; ++i) {
        if (w[i]<0 && -w[i]>-w[1]) ++cnt;
      }
    }
    }
```

```
        cout << cnt+1 << endl;
    return 0;
}
```

A Piece of Cake

map 套 pair 维护

判断一个草莓在哪一块蛋糕上, 这个过程可以用 二分查找 logn 来实现, 然后用 map 套 pair 记录这块蛋糕上草莓数量 +1

```
#include <bits/stdc++.h>
#define x first
#define y second

using namespace std;

typedef pair<int,int> PII;
const int maxn = 2e5 + 5;
int a[maxn], b[maxn];
struct node {
  int x, y;
} w[maxn];

int main()
{
  int W, H; cin >> W >> H;
  int T; cin >> T;
  for (int i = 1; i <= T; ++i) cin >> w[i].x >> w[i].y;

  int n; cin >> n;
  for (int i = 1; i <= n; ++i) cin >> a[i];
  int m; cin >> m;
  for (int i = 1; i <= m; ++i) cin >> b[i];

  map<PII, int> mp;
  for (int i = 1; i <= T; ++i) {
    int x = w[i].x, y = w[i].y;
    int tx = lower_bound(a+1, a+n+1, x) - a, ty = lower_bound(b+1, b+m+1, y) - b;
    mp[{tx,ty}]++;
  }

  int minn = 1e9, maxx = 0;
  if (mp.size() != (n+1)*(m+1)) minn = 0;
  for (auto it : mp) minn = min(minn, it.y), maxx = max(maxx, it.y);
  cout << minn << " " << maxx << endl;
  return 0;
}
```