二维前缀和

人员

初锦阳、赵牧之、王馨琪、刘宸熙、苑钊、温郝冬、倪炜艺、柳力玮、田心一、姜皓轩、谢梓轩、李知朔、韩昱辰、栾婷婷、燕润石、宋吉相 到课, 刘子轩 线上

上周作业检查

上周作业链接: https://www.luogu.com.cn/contest/237833

2025-0323周日15:30

报名 编辑比赛 题目数 7 报名人数 25

比赛说明 题目列表 排行榜

名次	参赛者	总分	A	B	C	D	E	F	G
#1	柳力玮	608 (7.50h)	100 (14.20min)	100 (24.00min)	100 (33.88min)	100 (1.90h)	100 (57.13min)	8 (1.43h)	100 (2.01h)
#2	赵牧之	608 (10.92d)	100 (19.33min)	100 (45.65min)	100 (1.27d)	100 (2.02h)	100 (1.28d)	8 (3.22d)	100 (5.02d)
#3	苑钊	508 (15.39d)	100 (23.20min)	100 (49.32min)	100 (1.71h)	100 (6.89d)	100 (1.65d)	8 (6.72d)	
#4	刘宸熙	508 (18.05d)	100 (55.85min)	100 (1.64h)	100 (4.30d)	100 (4.30d)	100 (6.05d)	8 (3.29d)	
#5	田心一	500 (6.69h)	100 (25.50min)	100 (49.35min)	100 (1.59h)	100 (1.80h)	100 (2.05h)	0	
#6	李知朔	500 (12.58d)	100 (53.30min)	100 (53.08min)	100 (6.21d)	100 (1.83h)	100 (6.22d)		
#7	温郝冬	500 (12.62d)	100 (24.95min)	100 (55.38min)	100 (2.15h)	100 (6.23d)	100 (6.24d)		
#8	宋云朗	500 (35.23d)	100 (7.00d)	100 (7.04d)	100 (7.06d)	100 (7.06d)	100 (7.07d)		
#9	周沁言	500 (35.26d)	100 (7.03d)	100 (7.03d)	100 (7.06d)	100 (7.07d)	100 (7.07d)		
#10	王馨琪	450 (13.67h)	100 (54.58min)	100 (59.97min)	80 (1.58h)	100 (4.96h)	70 (5.22h)		
#11	谢梓轩	400 (5.04h)	100 (27.70min)	100 (58.95min)	100 (1.78h)	100 (1.82h)			
#12	姜皓轩	400 (7.41h)	100 (58.42min)	100 (2.35h)	100 (2.28h)	100 (1.81h)			
#13	初锦阳	400 (4.38d)	100 (32.55min)	100 (52.25min)	100 (1.71h)	100 (4.25d)		0	
#14	韩昱辰	400 (27.86d)	100 (6.95d)	100 (6.95d)	100 (6.98d)	100 (6.99d)			
#15	纪博涵	310 (4.16h)	100 (28.07min)	100 (43.27min)	10 (55.03min)	100 (2.05h)			
#16	倪炜艺	300 (2.47h)	100 (24.57min)	100 (52.40min)	100 (1.19h)				
#17	李瑞涵	270 (2.61h)	100 (34.03min)	100 (58.20min)	70 (1.07h)				
#18	栾婷婷	200 (12.03d)	100 (6.02d)		100 (6.01d)				
#19	燕润石	140 (13.75d)	100 (6.87d)			40 (6.88d)			
#20	半仙胡小桃	100 (5.05d)					100 (5.05d)		

作业

https://www.luogu.com.cn/contest/238527 (课上讲了 A ~ D 题, 课后必做作业是 E 题, 选做作业是 F 题)

课堂表现

今天学的二维前缀和, 内容不是很难, 大家课下一定要多画几遍图, 掌握二维前缀和是怎么推出来的。

课堂内容

P6568 [NOI Online #3 提高组] 水壶

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

using namespace std;

const int maxn = 1e6 + 5;
int w[maxn], p[maxn];

int get_sum(int l, int r) { return p[r] - p[l-1]; }

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

  int res = 0;
  for (int i = 1; i+k-1 <= n; ++i) {
    res = max(res, get_sum(i, i+k-1));
  }
  cout << res << endl;
  return 0;
}
```

P10233 [yLCPC2024] A. dx 分计算

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

using namespace std;

const int maxn = 1e7 + 5;
char s[maxn];
int w[maxn], p[maxn];

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

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

```
  for (int i = 1; i <= n; ++i) {
    if (s[i] == 'P') w[i] = 3;
    else if (s[i] == 'p') w[i] = 2;
    else if (s[i] == 'G') w[i] = 1;
    else w[i] = 0;

    p[i] = p[i-1] + w[i];
  }

  int m; cin >> m;
  while (m -- ) {
    int l, r; cin >> l >> r;
    cout << get_sum(l, r) << endl;
  }
}

int main()
{
  int T; cin >> T;
  while (T -- ) solve();
  return 0;
}
```

P10416 [蓝桥杯 2023 国 A] XYZ

当 Z == 2*L 时, X/Y 的方案只有 1 种

当 Z == 2*L+1 时, X/Y 的方案有 2 种

当 Z == 2*L+2 时, X/Y 的方案有 3 种

...

当 Z == R 时, X/Y 的方案有 R-2*L+1 种

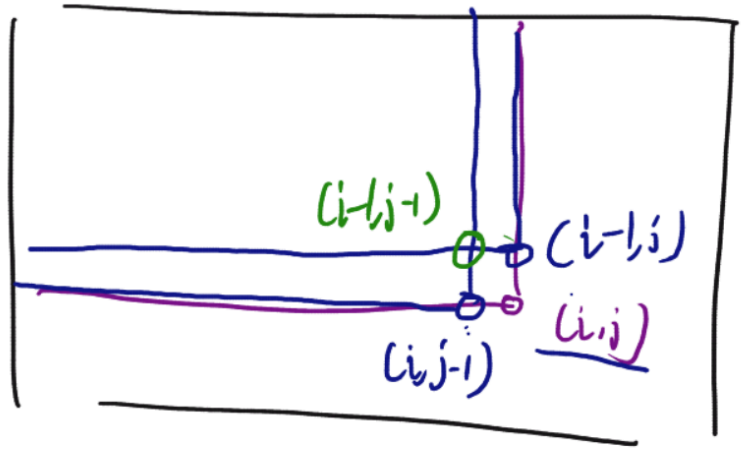
所以, 总答案应该是 1 + 2 + 3 + ... + R-2*L+1, 可以用等差数列 O(1) 求和

需要注意的问题: R-2*L+1 有可能 小于1, 此时说明无解, 应该输出 0

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

using namespace std;

typedef long long LL;

LL get_sum(int l, int r) {
  if (l > r) return 0;
  return ((LL)l+r)*(r-l+1)/2;
}

int main()
{
```

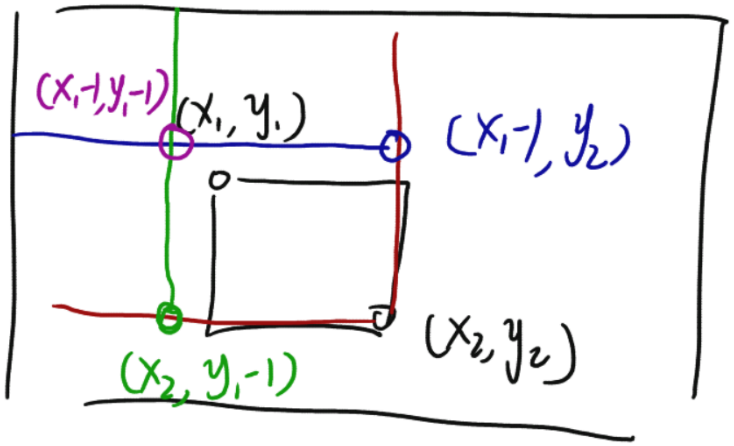
```
  int T; cin >> T;
  while (T -- ) {
    int l, r; cin >> l >> r;
    cout << get_sum(1, r-2*l+1) << endl;
  }
  return 0;
}
```

P2004 领地选择



$$P[i][j] = P[i-1][j] + P[i][j-1] - P[i-1][j-1] + a[i][j]$$

$$P[x_2][y_2] - P[x_1-1][y_2] - P[x_2][y_1-1] + P[x_1-1][y_1-1]$$




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

using namespace std;
```

```
const int maxn = 1e3 + 5;
int a[maxn][maxn], p[maxn][maxn];

int calc(int x1, int y1, int x2, int y2) {
    return p[x2][y2] - p[x1-1][y2] - p[x2][y1-1] + p[x1-1][y1-1];
}

int main()
{
    int n, m, c; cin >> n >> m >> c;
    for (int i = 1; i <= n; i++) {
        for (int j = 1; j <= m; j++) cin >> a[i][j], p[i][j] = p[i-1][j] + p[i][j-
1] - p[i-1][j-1] + a[i][j];
    }

    int maxx = -1e9, x = 0, y = 0;
    for (int i = 1; i <= n-c+1; i++) {
        for (int j = 1; j <= m-c+1; j++) {
            int sum = calc(i,j,i+c-1,j+c-1);
            if (sum > maxx) {
                maxx = sum, x = i, y = j;
            }
        }
    }
    cout << x << " " << y << endl;
    return 0;
}
```

P1719 最大加权矩形

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

using namespace std;

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

int get_sum(int x1, int y1, int x2, int y2) {
  return p[x2][y2] - p[x1-1][y2] - p[x2][y1-1] + p[x1-1][y1-1];
}

int main()
{
  int n; cin >> n;
  for (int i = 1; i <= n; ++i) {
    for (int j = 1; j <= n; ++j) {
      int x; cin >> x; p[i][j] = p[i-1][j] + p[i][j-1] - p[i-1][j-1] + x;
    }
  }
```

```
  int maxx = -1e9;
  for (int i1 = 1; i1 <= n; ++i1) {
    for (int j1 = 1; j1 <= n; ++j1) {
      for (int i2 = i1; i2 <= n; ++i2) {
        for (int j2 = j1; j2 <= n; ++j2) maxx = max(maxx, get_sum(i1,j1,i2,j2));
      }
    }
  }
  cout << maxx << endl;
  return 0;
}
```