

- [问题 A: 小G的日常之折扣票价](#)
- [问题 B: 画箭](#)
- [问题 C: FJ的字符串](#)
- [问题 D: 抢机位](#)
- [问题 E: 你爱我么?](#)
- [问题 F: 小明发福利](#)
- [问题 G: acm可好玩了](#)
- [问题 H: 输出直角梯形](#)
- [问题 I: 学长的那些事系列之--暑期集训](#)
- [问题 J: 晓美焰の忧郁](#)

问题 A: 小G的日常之折扣票价

第一次原价，第二次除2，然后加起来出答案

```
#include <stdio.h>

int main () {
    int x, y; scanf("%d%d", &x, &y);
    printf("%.2f", x * 1.0 + y * 1.0 / 2);
}
```

问题 B: 画箭

先输出三个'>'
然后输出n个'-'
最后输出一个'>'

```
#include <stdio.h>

int main () {
    int n; scanf("%d", &n);
    printf(">>>");
    while ( n -- ) printf("-");
    printf(">");
}
```

问题 C: FJ的字符串

(从0开始)
规律：每进一步，相当于原字符串+('A'+进的步数+1)+原字符串
那么就每次按这个规律进行模拟就行了
最多 $(2+1) \times 26$ 次，就是67108863，这就是初始化的size

```
#include <stdio.h>
#include <string.h>

char s[67108866];
int cnt = 0; // 字符串的size

int main () {
    int n; scanf("%d", &n);
    for ( int add = 0; add < n; add ++ ) {
        s[cnt] = 'A' + add;
        for ( int i = 0; i < cnt; i ++ ) s[i + cnt + 1] = s[i];
        cnt = cnt * 2 + 1;
    }
    printf("%s", s);
}
```

问题 D: 抢机位

首先就是读入两个数组代表物工和通信每个时间点来的人数
设置suma,sumb代表两个专业的已拿机位数
然后对于每个时间点进行判断
如果剩余机位够的话就两个都加上
否则进行判断，专业最多的可以选择min(当前时间点人数，剩余机位数量一半上取整个)
最少的拿走剩下的
最后进行判断suma和sumb的关系

```
#include <stdio.h>
int Min ( int x, int y ) { return x > y ? y : x; }

int main () {
    int m, n; scanf("%d%d", &m, &n);
    int a[n], b[n];
```

```

for ( int i = 0; i < n; i ++ ) scanf("%d", &a[i]);
for ( int i = 0; i < n; i ++ ) scanf("%d", &b[i]);
int suma = 0, sumb = 0;
for ( int i = 0; i < n; i ++ ) {
    if ( m - a[i] - b[i] >= 0 ) {
        suma += a[i],
        sumb += b[i];
    } else {
        if ( a[i] >= b[i] ) {
            int aget = Min ( m / 2 + (m & 1), a[i] );
            int bget = m - aget;
            suma += aget, sumb += bget;
        } else {
            int bget = Min ( m / 2 + (m & 1), b[i] );
            int aget = m - bget;
            suma += aget, sumb += bget;
        }
        break;
    }
    m -= a[i] + b[i];
}
if ( suma > sumb ) printf("物工胜");
else if ( suma == sumb ) printf("平局");
else printf("通信胜");
}

```

问题 E: 你爱我么？

分析一下题目，就是让你凑出来最大的奇数

那么对奇数存一个数组

偶数一定要拿，然后拿从奇数数组内拿奇数个最大的数即可

```

#include <stdio.h>

int main () {
    int cass;
    for ( scanf("%d", &cass); cass; cass -- ) {
        int n; scanf("%d", &n);
        int odd[n], cntodd = 0;
        int res = 0;
        for ( int i = 0, x; i < n; i ++ ) {
            scanf("%d", &x);
            if ( x & 1 ) odd[cntodd++] = x;
            else res += x;
        }
        // 排序
        for ( int i = 0; i < cntodd; i ++ )
            for ( int j = i + 1; j < cntodd; j ++ )
                if ( odd[j] > odd[i] ) {
                    int tmp = odd[j];
                    odd[j] = odd[i];
                    odd[i] = tmp;
                }
        if ( cntodd == 0 ) {
            printf("0\n");
        } else {
            res += odd[0];
            for ( int i = 1; i < cntodd - 1; i += 2 )
                res += odd[i],
                res += odd[i + 1];
            printf("%d\n", res);
        }
    }
}

```

问题 F: 小明发福利

超过总和，意味着拿到的糖的个数>n/2，所以就设置一个数组

每次对输入x的vis[x]++

每次判断一下，如果有vis[x]>n/2那么就是这个人要惩罚

```

#include <stdio.h>

int main () {
    int n; while ( scanf("%d", &n) != EOF ) {
        int vis[11000] = {0}, res = -1;
        for ( int i = 0, x; i < n; i ++ ) {
            scanf("%d", &x); vis[x] ++;
            if ( vis[x] > n / 2 && res == -1 ) res = x;
        }
        printf("%d\n", res);
    }
}

```

问题 G: acm可好玩了

设置两个下标l,r来截取我们当前的字符串，递进r
如果s[r]是空格，就将这一部分截取的字符串倒着输出
然后l=r+1
(小简单操作，最后一个设置为空格，可以避免很多麻烦)

```
#include <stdio.h>
#include <string.h>

char s[5050];

int main () {
    gets(s);
    int len = strlen(s);
    s[len] = ' ';
    for ( int l = 0, r = 0; r <= len; r ++ ) {
        if ( s[r] == ' ' ) {
            for ( int i = r - 1; i >= l; i -- ) printf("%c", s[i]);
            printf(" ");
            l = r + 1;
        }
    }
}
```

问题 H: 输出直角梯形

比赛呢，顾及那么多要求干嘛
直接循环多香，每一行的空格数和星星数都有规律

```
#include <stdio.h>

void printTrapezoid(int rows,int cols) {
    for ( int i = 0; i < rows; i ++ ) {
        for ( int j = 0; j < rows - i - 1; j ++ ) printf(" ");
        for ( int j = rows - i - 1; j < cols; j ++ ) printf("*");
        printf("\n");
    }
}

int main () {
    int n, m; scanf("%d%d", &n, &m);
    printTrapezoid(n, m);
}
```

问题 I: 学长的那些事系列之--暑期集训

每次比赛每个人按成绩大小离散化为自己的名次（使用结构体）
然后对每个人的总成绩加上自己的名次
最后再对每个人的总成绩按大小离散化为自己的名次
两次离散化都要细心，很麻烦，这是C语言的代码

```
#include <stdio.h>

typedef struct compete { int id, grade, ID; } cpt;
typedef struct student { int id, grade, ID; } stu;

cpt c[100];
stu s[100];
int m, n;

void get_sort_cpt () {
    for ( int i = 0; i < m; i ++ )
        for ( int j = i + 1; j < m; j ++ )
            if ( c[i].grade < c[j].grade ) {
                cpt tmp = c[i];
                c[i] = c[j];
                c[j] = tmp;
            }
    for ( int i = 0; i < m; i ++ ) {
        if ( i ) {
            if ( c[i].grade == c[i - 1].grade ) c[i].ID = c[i - 1].ID;
            else c[i].ID = i + 1;
        } else c[i].ID = 1;
    }
    for ( int i = 0; i < m; i ++ )
        for ( int j = i + 1; j < m; j ++ )
            if ( c[i].id > c[j].id ) {
                cpt tmp = c[i];
                c[i] = c[j];
                c[j] = tmp;
            }
}
```

```

void get_sort_stu () {
    for ( int i = 0; i < m; i ++ ) s[i].id = i;
    for ( int i = 0; i < m; i ++ )
        for ( int j = i + 1; j < m; j ++ )
            if ( s[i].grade > s[j].grade ) {
                stu tmp = s[j];
                s[j] = s[i];
                s[i] = tmp;
            }
    for ( int i = 0; i < m; i ++ ) {
        if ( i ) {
            if ( s[i].grade == s[i - 1].grade ) s[i].ID = s[i - 1].ID;
            else s[i].ID = i + 1;
        }
        else s[i].ID = 1;
    }
    for ( int i = 0; i < m; i ++ )
        for ( int j = i + 1; j < m; j ++ )
            if ( s[i].id > s[j].id ) {
                stu tmp = s[j];
                s[j] = s[i];
                s[i] = tmp;
            }
}

int main () {
    scanf("%d%d", &m, &n);
    for ( int i = 0; i < n; i ++ ) {
        for ( int j = 0; j < m; j ++ )
            scanf("%d", &c[j].grade),
            c[j].id = j;
        get_sort_cpt();
        for ( int j = 0; j < m; j ++ )
            s[j].grade += c[j].ID;
    }
    get_sort_stu();
    for ( int i = 0; i < m; i ++ ) printf("%d ", s[i].ID);
}

```

然后我们使用C++进行离散化就会很方便，存到vector内sort排序后，二分出来第一个大于等于这个数的数的位置进行离散化

```

#include <iostream>
#include <vector>
#include <algorithm>

using namespace std;

int m, n;
int stu[100];
vector<int> nums;

int main () {
    scanf("%d%d", &m, &n);
    for ( int i = 0; i < n; i ++ ) {
        nums.clear(); int a[100];
        for ( int j = 0; j < m; j ++ )
            cin >> a[j],
            nums.push_back (a[j]);
        sort ( nums.begin(), nums.end(), greater<int>());
        for ( int j = 0; j < m; j ++ )
            stu[j] += lower_bound(nums.begin(), nums.end(), a[j], greater<int>()) - nums.begin() + 1;
    }
    nums.clear();
    for ( int i = 0; i < m; i ++ ) nums.push_back(stu[i]);
    sort ( nums.begin(), nums.end() );
    for ( int i = 0; i < m; i ++ )
        cout << lower_bound(nums.begin(), nums.end(), stu[i]) - nums.begin() + 1 << " ";
}

```

问题 J: 晓美焰の忧郁

		粉	白	粉	粉	白	粉		
			粉	青	粉	粉	青	粉	
				粉	白	粉	粉	白	粉
					粉	青	粉	粉	青
						粉	白	粉	粉
							粉	青	粉
								粉	白
									粉

可以发现这样构造时每3列就有两列是粉

所以输出 $\frac{2}{3}$ 即可

double存不下，可以手动输出0.666666...67

```
#include <stdio.h>
int main () {
    printf("0.");
    for ( int i = 0; i < 19; i ++ ) printf("6");
    printf("7");
}
```