



中北大学 ACM-ICPC 2014 年 新生赛题解

Edit By Hu Jie
2014.12.22
1:00am

(配合新生赛题目食用风味更佳)

Problem List

Problem A : 军训方阵

Problem B : 礼物包装

Problem C : 进制转换

Problem D : 女神的任务 I

Problem E : 女神的任务 II

Problem F : 座位距离

Problem G : 情书

Problem H : 成绩

Problem I : 汉诺塔++

Problem J : 浪漫的表白

Problem A : 军训方阵

题意是给一个数 n , 打印出 $a * b == n$ 中的所有 $a * b$ 。

从 1 枚举到 \sqrt{n} , 一一打印即可。

PE 的坑点在于空格。刚开始的时候 PE 了两发。T T

Code:

```
#include <cstdio>
#include <cmath>

int main()
{
    //freopen( "A.in", "r", stdin );
    //freopen( "A.out", "w", stdout );

    int n, cas = 1;
    while ( scanf( "%d", &n ) != EOF )
    {
        printf( "Case %d:\n", cas++ );
        int end = sqrt(n);
        for ( int i=1; i<=end; i++ )
            if ( n % i == 0 )
                printf( "%d * %d\n", i, n/i );
        printf( "\n" );
    }
    return 0;
}
```

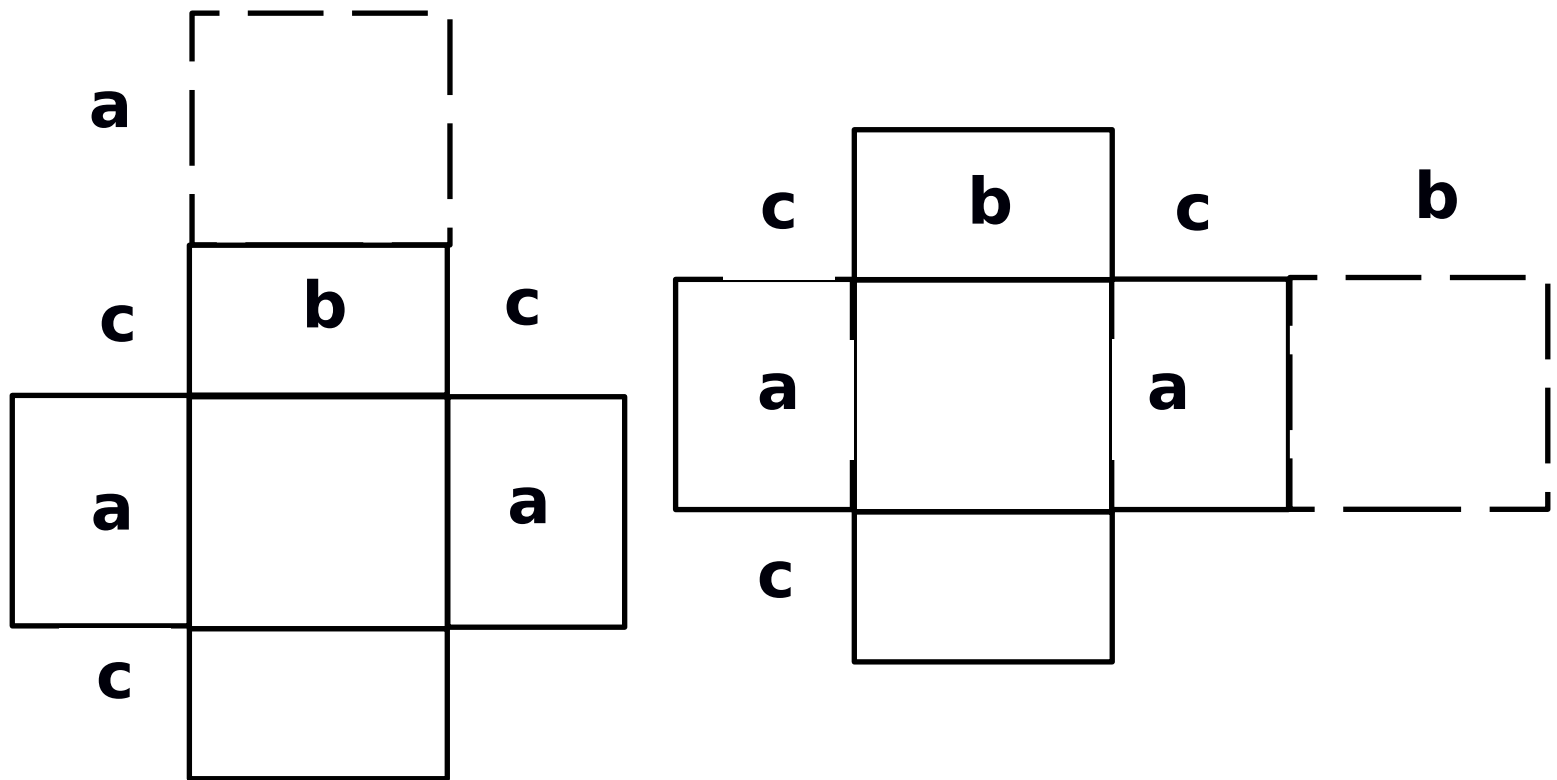
Problem B : 礼物包装

题意是给出一个长方体的长宽高 a, b, c , 求一个不能裁剪的矩形包装纸, 要求其面积是最小的。

把所有情况列举出来随便搞一搞就行了。可以分别按三个不同的面打开, 每个不同打开面又有顶盖加在长上或者宽上两种情况。

做的时候挫逼卡题了, 用 LL 搞了半天 min 函数调用不出来 balabalablab.....

用 ab 面打开的两种情况如下:



$$\begin{aligned}\text{对应的表达式为: } \text{area} &= (2 * a + 2 * c) * (b + 2 * c) \\ \text{area} &= (a + 2 * c) * (2 * b + 2 * c)\end{aligned}$$

Code:

标程序好像出了点问题，所以贴了我比赛时的代码：

```
#include <cstdio>
#include <cstring>
#include <cstdlib>
#include <iostream>
#include <algorithm>
#include <string>
#include <vector>
#include <cmath>
#define LL long long
using namespace std;

const int maxn = 1e6;

int main()
{
#ifdef LOCAL
    freopen("in.txt", "r", stdin);
#endif // LOCAL
    int a, b, c;
    int ans;
    int area;
    while (scanf("%d %d %d", &a, &b, &c) == 3)
    {
        ans = 1000000000 + 10;
        area = 0;
        area = (2 * a + 2 * c) * (b + 2 * c);
        if (ans > area)
        {
            ans = area;
        }

        area = (a + 2 * c) * (2 * b + 2 * c);
        if (ans > area)
        {
            ans = area;
        }

        area = (2 * b + 2 * a) * (c + 2 * a);
        if (ans > area)
        {
            ans = area;
        }

        area = (b + 2 * a) * (2 * c + 2 * a);
        if (ans > area)
        {
            ans = area;
        }
    }
}
```

```
    area = (2 * a + 2 * b) * (c + 2 * b);  
    if (ans > area)  
    {  
        ans = area;  
    }  
  
    area = (a + 2 * b) * (2 * c + 2 * b);  
    if (ans > area)  
    {  
        ans = area;  
    }  
    printf("%d\n", ans);  
}  
return 0;  
}
```

Problem C : 进制转换

题意是给一个字符串 n ，两个整数 s, t ，要求把 s 进制下的数 n 转换成 t 进制下的数输出。

按照二进制的转换想法去转换就行了，比如 ffff 16 7，先把 ffff 转换成十进制下的数 $sum = f * 16^3 + f * 16^2 + f * 16^1 + f * 16^0$ 。

然后再不断保存 $sum \% 7$ 并 $sum /= 7$ 来将转换进制后的数字保存起来，最后倒序输出就行了（想想为什么要倒序输出）。关于字母的处理方法，只要大于等于 10 到 35 的数都可以用字母表示，比如 $c = B$ 代表 11，只要 $c - 'A' + 10$ ，就可以顺利转换成它所代表的数字了喵。

题目的坑点一在于编码习惯，在人工组建字符串的时候，一定要在字符串末尾加 `'\0'`；

坑点二在于输入可能包含大写小写字母，最后的输出是大写。

Code:

标程：

```
#include <cstdio>
#include <cstring>

const int length = 32;

inline int Num( char c )
{
    if ( c < 'A' )
        return ( c & 0x0f );
    return ( ( c & 0x1f ) + 9 );
}

inline int Chr( int x )
{
    if ( x < 10 )
        return ( x + '0' );
    return ( x - 10 + 'A' );
}

int Str_to_Num( char *src, int m )
{
    int ret = 0;
    while ( *src )
        ret = ret * m + Num( *src++ );
    return ret;
}

char *Num_to_Str( int n, int m, char *dst )
{
```

```

if ( n == 0 )
{
    dst[0] = '0';
    dst[1] = '\0';
    return dst;
}
int i = length - 1;
dst[i] = '\0';
while ( n && (--i) )
{
    dst[i] = Chr( n % m );
    n /= m;
}
return ( dst + i );
}

int main()
{
    //freopen( "E.in", "r", stdin );
    //freopen( "E.out", "w", stdout );

    char str[length];
    int a, b;
    while ( scanf( "%s%d%d", str, &a, &b ) )
    {
        if ( ( a | b ) == 0 )
            break;
        printf( "%s\n", Num_to_Str( Str_to_Num( str, a ), b, str ) );
    }

    return 0;
}

```

以下是我比赛时的代码：

```

#include <cstdio>
#include <cstring>
#include <cstdlib>
#include <iostream>
#include <algorithm>
#include <cmath>
#define LL long long
using namespace std;

const int maxn = 10 + 10;

char s[maxn];
char ans[maxn];
int fr, to;

int fun(int f, int time)

```



```

{
    int res = 1;
    for (int i = 1; i <= time; i++)
    {
        res *= f;
    }
    return res;
}

void change(int f, int t)
{
    int sum = 0;
    int len = strlen(s);
    for (int i = len - 1; i >= 0; i--)
    {
        int t = 0;
        if ('0' <= s[i] && s[i] <= '9')
        {
            sum += fun(f, len - i - 1) * (s[i] - '0');
        }
        if ('A' <= s[i] && s[i] <= 'Z')
        {
            sum += fun(f, len - i - 1) * (s[i] - 'A' + 10);
        }
        if ('a' <= s[i] && s[i] <= 'z')
        {
            sum += fun(f, len - i - 1) * (s[i] - 'a' + 10);
        }
    }
    int jj = 0;
    while (sum)
    {
        int d = sum % t;
        if (d >= 10)
        {
            ans[jj] = 'A' + d - 10;
            jj++;
        }
        else
        {
            ans[jj] = '0' + d;
            jj++;
        }
        sum /= t;
    }
    ans[jj] = '\0';///就是这个地方....T T
}

int main()
{

```

```
#ifdef LOCAL
freopen("in.txt", "r", stdin);
#endif // LOCAL
while (scanf("%s%d%d", s, &fr, &to) == 3)
{
    getchar();
    if (s[0] == '0' && fr == 0 && to == 0)
        break;
    memset(ans, 0, sizeof(ans));
    change(fr, to);
    int len = strlen(ans);
    for (int i = len - 1; i >= 0; i--)
    {
        printf("%c", ans[i]);
    }
    printf("\n");
    memset(s, 0, sizeof(s));
}
return 0;
}
```

Problem D : 女神的任务 I

题意是给两个整数 n, m 代表人数和考核的项目数，然后给出 m 个整数 ($i=1..M$)，表示要达到“良”等级时，第 i 项的最低要求分数，之后 N 行每行 M 个整数 T_i ($i=1..M$)，表示实验室成员第 i 项考核成绩。求每一个考核项目都过关的人的总数。

题意读懂就 k.o 了。（中文题就不说什么了）。

Code:

```
#include <cstdio>
int main()
{
    int n, m, sum, score[11], tmp;
    bool flag;

    while ( scanf( "%d%d", &n, &m ) != EOF )
    {
        if ( ( n | m ) == 0 )
            break;
        for ( int i=0; i<m; i++ )
            scanf( "%d", score+i );
        sum = 0;
        for ( int i=0; i<n; i++ )
        {
            flag = true;
            for ( int j=0; j<m; j++ )
            {
                scanf( "%d", &tmp );
                if ( flag && ( tmp < score[j] ) )
                    flag = false;
            }
            sum += ( flag ? 1 : 0 );
        }
        printf( "%d\n", sum );
    }
    return 0;
}
```

Problem E : 女神的任务 II

给 n, m, s , 表示班级人数, 学分相关项目种类, 班级名称。然后给出 m 个项目的学分 $score[i]$ 。求这个班级获得的总学分。

$$scoreSum = n * \sum_{i=1}^m score[i]$$

Code:

```
#include <cstdio>
#include <cstdlib>

int main()
{
    //freopen( "C.in", "r", stdin );
    //freopen( "C.out", "w", stdout );

    int t, n, m;
    double tmp, sum;
    char name[24];

    scanf( "%d", &t );
    while ( t-- )
    {
        scanf( "%d%d%s", &n, &m, name );
        sum = 0.0;
        for ( int i=0; i<m; i++ )
        {
            scanf( "%lf", &tmp );
            sum += tmp;
        }
        printf( "The scores of Class %s's is %.2lf.\n", name, sum*n );
    }
    return 0;
}
```

Problem F : 座位距离

题意是给定一个 $n*m$ 的图，求 'G' 点（总觉得哪里怪怪的=￣ω￣=）到 '.' 间的最小哈密顿距离，把哈密瓜距离搞懂对于数据量并不大的这一题就好解决了。

——枚举每一个点，若不存在 '.'，就输出那个 sorry 什么的，如果存在，——去比较久好了。

一个学弟赛后问我说他图一直读不进去，我说因为他没有 getchar()。（因为他没有摩擦摩擦...）。getchar()用来吃掉输入中的回车，比如输完第一行，会输入一个回车，这样就要用到 getchar()。

Code:

标程：

```
#include <cstdio>
#include <cstring>

#define is_godnes=='G'
#define is_empty=='.'
#define is_occupied=='#'

#define ergodic_x(xx) for(int i=0;i<n;i++){xx}
#define ergodic_y(yy) for(int j=0;j<m;j++){yy}
#define abs(x) ((x<0)?(-(x)):(x))

#define BIG 99999999

int main()
{
    //freopen( "F.in", "r", stdin );
    //freopen( "F.out", "w", stdout );

    int n, m, x, y, ans;
    int tx, ty;
    char f[102][102];
    while ( scanf( "%d%d", &n, &m ) != EOF )
    {
        if ( ( n | m ) == 0 )break;
        ans = BIG;
        ergodic_x( scanf( "%s", f[i] ); )
        ergodic_x( ergodic_y( if( f[i][j] is_godnes )
        {
            x=i;
            y=j;
        } ) )
        ergodic_x( ergodic_y( if( f[i][j] is_empty )
        {
```

```

        if( abs(i-x)+abs(j-y)<ans )
        {
            ans=abs(i-x)+abs(j-y);
            tx=i;
            ty=j;
        }
    } ) )
    if ( ans >= BIG )printf( "Sorry, there are no more seats.\n" );
    else printf( "Hurry, the shortest distance between you and
godness is %d.\n", ans );
}

return 0;
}

```

比赛时候我的代码：

```

#include <cstdio>
#include <cstring>
#include <cstdlib>
#include <iostream>
#include <algorithm>
#include <string>
#include <vector>
#include <cmath>
#define LL long long
using namespace std;

const int maxn = 1e6;

char g[200][200];

int main()
{
#ifdef LOCAL
    freopen("in.txt", "r", stdin);
#endif // LOCAL
    int n, m;
    while (scanf("%d%d", &n, &m) == 2)
    {
        memset(g, 0, sizeof(g));
        if (n == 0 && m == 0)
            break;
        int posx = -1, posy = -1;
        getchar();
        for (int i = 0; i < n; i++)
        {
            for (int j = 0; j < m; j++)
            {
                scanf("%c", &g[i][j]);
                if (g[i][j] == 'G')

```

```

        {
            posx = i;
            posy = j;
        }
    }
    getchar();
}
bool flag = false;
int ans = 1000000;
for (int i = 0; i < n; i++)
{
    for (int j = 0; j < m; j++)
    {
        if (g[i][j] == '.')
        {
            int sum = abs(posx - i) + abs(posy - j);
            if (sum < ans)
                ans = sum;
            flag = true;
        }
    }
}
if (!flag)
{
    printf("Sorry, there are no more seats.\n");
}
else
{
    printf("Hurry, the shortest distance between you and
godness is %d.\n", ans);
}
return 0;
}

```

Problem G : 情书

两道没有 A 出来的题之一，按照平常想法去做没办法不是 RE 就是 TLE。简直悲伤逆流成河。Ps.Love Letter 这首歌不错。

题意是给好多个字符串，把这些字符串倒序输出。

我的做法是一个一个单字符读入，遇到空格就可以进入下一个字符串的保存了，遇到'\n'就可以 break 了。就是这种做法坑了，原因是输入的时候有可能有很多个空格，这样的话在遇到空格的时候就自动进入下一个字符串，不断的往后推而字符串里面没有保存到任何东西。大概改进就是当遇到空格后，再次遇到一个字母时再进入下一个字符串的存储。说实话这题没 A 出来挺遗憾的。

标程的做法是将每一个字符串的首字母与末字母的下标记录下来最后按每一个末字母的下标来输出，最经典的地方在于输出那里的反斜杠 0 : '\0'还有字符串指针的输出控制。

Code:

```
#include<cstdio>
#include<cstring>
#include<iostream>
#include<algorithm>
using namespace std;
const int MAXN = 2e6;
const int MAXM = 2e5 + 10;
char str[MAXN];
int start[MAXM], End[MAXM];

int main(void)
{
#ifdef LOCAL
    freopen("data.in", "r", stdin);
    freopen("data.out", "w", stdout);
#endif
    int ncase;
    scanf("%d", &ncase);
    getchar();
    for(int ca = 1; ca <= ncase; ++ca)
    {
        gets(str);
        int len = strlen(str), n = 0;

        for(int i = 0; i < len; ++i)
            if(str[i] != ' ')
            {
                start[n] = i;
                while(str[i] != ' ' && i < len)
                    i++;
            }
    }
}
```



```
        End[n++] = i;
    }
    printf("Case #%%d:", ca);
    for(int i = n - 1; i >= 0; i--)
    {
        char c = str[End[i]];
        str[End[i]] = '\0';
        printf(" %s", str + start[i]);
        str[End[i]] = c;
    }
    putchar(10);
}
return 0;
}
```

Problem H : 成绩

题意是给两个数 n 和 m ， n 表示学生人数，给出每个学生的分数； m 表示询问次数，每次询问给出两个整数 L, R ，代表要查询的成绩范围。最后要求输出有多少个同学成绩在 $[L, R]$ 区间。由于 $n*m$ 的规模很大，所以裸上一个二分，求出上下界的下标，相减就是结果。注意 $L > R$ 的输出就行了。

具体解释：先把每个同学的成绩按从小到大排列，（有序是二分的前提），然后对每个区间 $[L, R]$ ，用二分法求下界找到不小于 L 的分数的最小下标，同理用二分法求上届找到不大于 R 的最大下标，两个下标相减就是这之间有多少个分数了。

Code:

```
#include <cstdio>
#include <cstring>
#include <iostream>
#include <algorithm>
using namespace std;
const int MAXN=1e6;

int a[MAXN+50];

int main()
{
    //freopen("data.in","r",stdin);
    // freopen("data.out","w",stdout);
    int T;
    scanf("%d",&T);
    while (T--)
    {
        int n,m;
        scanf("%d %d",&n,&m);
        for (int i=0; i!=n; i++)
            scanf("%d",&a[i]);
        sort(a,a+n);

        while (m--)
        {
            int s,t;
            scanf("%d %d",&s,&t);
            if (s>t)
            {
                printf("%d\n",-1);
                continue;
            }
            int pos1=lower_bound(a,a+n,s)-a;
            int pos2=upper_bound(a,a+n,t)-a;
```

```
        printf("%d\n",pos2-pos1);
    }
}
return 0;
}
```

Problem I : 汉诺塔++

这是这次新生赛最爱的一道题目。

题意就是一个 update 版的汉诺塔问题。

首先先普及一下 normal 版的汉诺塔问题：给你三根柱子 A, B, C, 把在柱子 A 上的 n 个从大到小向上叠着的圆环移动到 C, 要求移动过程中小的圆环要在大的圆环上面, 最后移动到 C 的状态时也要从大到小向上相叠。这是一个很简单的递归, 递归基是当仅有一个圆环时直接把这个环移动到 C, 而当有 n 个圆环时先借助 C 把 $n - 1$ 个环移动到 B 上, 然后把最下的一个环移动到 C 上, 最后再借助 A 把 $n - 1$ 个环移动到 C 上即可。

Python 代码实现 normal 汉诺塔：

```
def move(n, a, b, c):
    if n == 1:
        print a + '->' + c
    else:
        move(n - 1, a, c, b)
        move(1, a, b, c)      #此处伏笔
        move(n - 1, b, a, c)
```

然后到了 update 版, 意思是每次的移动只能是相邻的, 所以递归就变成了如下过程：递归基是当仅有一个圆环时, 采用穿越的方法, 从 A 移到 B, 再从 B 移到 C, 而当有 n 个圆环时, 先要把 $n - 1$ 个圆环穿越到 C, 然后把最下的一个圆环移动到 B, 然后让 $n - 1$ 个圆环穿越回 A, B 上的圆环移动到 C, A 上的 $n - 1$ 个圆环穿越到 C 就好了。开始 wa 了两发是因为上面的那句伏笔, 我写递归基的时候喜欢直接调用, 而 update 版的直接调用会把最下的那个移动的圆环穿越的乱七八糟, 直接在这里写输出就好。

看不懂思路可以拿纸笔画一画, 立马就清晰了。

Code:

```
#include <cstdio>

void Hanoi( char A, char B, char C, int n )
{
    if ( n <= 1 )
    {
        printf( "%c->%c\n", A, B );
        printf( "%c->%c\n", B, C );
        return;
    }
    Hanoi( A, B, C, n-1 );
    printf( "%c->%c\n", A, B );
    Hanoi( C, B, A, n-1 );
    printf( "%c->%c\n", B, C );
}
```

```
    Hanoi( A, B, C, n-1 );
}

int main()
{
    //freopen( "l.in", "r", stdin );
    //freopen( "l.out", "w", stdout );

    int n, cas = 1;
    while ( scanf( "%d",&n ) != EOF )
    {
        printf( "Case %d:\n", cas++ );
        Hanoi( 'A', 'B', 'C', n );
        printf( "\n" );
    }
    return 0;
}
```

Problem J : 浪漫的表白

最后实在写不下去的一题。

找到规律然后乱搞一下就能打出来吧，还有一种 yy 出来的方法就是反正就 9 个心形，一个一个写出来就行了。这些解法都是我编的。

哈哈哈哈哈。

膜拜标程代码。

Code:

```
#include <cstdio>
#include <cstring>
#include <cstdlib>

#define MAX_N 56
#define MAX_M 56
#define draw; f[x][y]='#';ret++;
#define move(b,a,c);for(int i=(c);i>0;i--)
{x+=(a);y+=(b);draw;}//printf("[%d,%d]\n",x,y);
#define center 3*(n-1),3*(n-1)
char f[MAX_N][MAX_M];

void INI()
{
    memset( f, ' ', sizeof(f) );
}

int MAKE( int cx, int cy, int n )
{
    if ( n <= 1 )
        return 0;
    int ret = 0;
    int x = cx - 2*(n-1);
    int y = cy;
    move(1,-1,n-1);
    move(1,0,n-1);
    move(1,1,n-1);
    move(0,1,2*(n-1));
    move(-1,1,3*(n-1));
    move(-1,-1,3*(n-1));
    move(0,-1,2*(n-1));
    move(1,-1,n-1);
    move(1,0,n-1);
    move(1,1,n-1);
    ret += MAKE( cx, cy, n-1 );
    return ret;
}

int ENDLINE()
```

```

{
    int i, j;
    for ( i=0; i<MAX_N; i++ )
    {
        for ( j=MAX_M-2; j>=0; j-- )
        {
            if ( f[i][j] != ' ' )
            {
                f[i][j+1] = '\0';
                break;
            }
        }
        if ( j < 0 )
            break;
    }
    return i;
}

int main()
{
    //freopen( "Heart.in", "r", stdin );
    //freopen( "Heart.out", "w", stdout );
    int n;
    while ( scanf( "%d", &n ) != EOF )
    {
        INI();
        printf( "He needs %d candles.\n", MAKE( center, ++n ) );
        for ( int i=0, end=ENDLINE(); i<end; i++ )
            printf( "%s\n", f[i] );
        printf( "\n" );
    }
    return 0;
}

```

后记

感谢闫语畅、武宽、张亮、史可、李芳、王志强...还有分科协的学姐学长们出的题干和题面。Ps. 韬哥看完题目以后和我说，你可以给写题面发这个表情：



我要和你们同归于尽

（题面把我出的这么蠢基，我以后怎么找女朋友！）

愿 acm-icpc 实验室越办越好，热爱编程的人终有所获。

以上。