

BUAASOFT

第 3 次练习赛的解题报告

2014 级

14211045 张涛

2014/11/12

目录

A. 新安当的神笔..... 6

 题目描述: 6

 输入: 6

 输出: 6

 输入样例: 6

 输出样例: 6

 代码: 7

B 逃出迷宫 8

 Problem Description..... 8

 Input 8

 Output..... 8

 Sample Input..... 9

 Sample Output 9

 代码: 9

C 召唤英灵..... 10

 题目描述..... 10

 输入..... 10

 输出..... 10

 输入样例..... 10

 输出样例..... 11

代码:	11
D 盗墓笔记之禁婆的考验.....	12
题目描述.....	12
输入.....	12
输出.....	12
输入样例.....	12
输出样例.....	13
代码:	13
E 盗墓笔记之怒海潜沙.....	14
题目描述.....	14
输入.....	14
输出.....	14
输入样例.....	14
输出样例.....	14
HINT	15
代码:	15
F 盗墓笔记之逃亡.....	16
题目描述.....	16
输入.....	17
输出.....	17
输入样例.....	17

输出样例.....	17
代码:	17
G 李逍遥的仙剑客栈	18
Description.....	18
Input	18
Output.....	19
Sample.....	19
Hint	19
代码:	19
H jhljx 学函数	21
Problem Description.....	21
Input	21
Output.....	21
Sample Input.....	21
Sample Output	21
Hint	21
方法 1	22
方法 2	22
值传递.....	22
引用传递.....	23
代码 1:	23

代码 2:	24
I Ryan's Derivatives.....	25
Problem Description.....	25
Input	25
Output.....	25
Sample Input.....	26
Sample Output	26
代码:	27
J RecTangles! RecTangles! RecTangles!	28
题目描述.....	28
输入.....	28
输出.....	29
输入样例.....	29
输出样例.....	29
HINT	29
代码:	29

A. 新安当的神笔

题目描述：

新安当这天来了个稀客，没错就是传说中没有节操水平超越了延帝的天臧散人，Arthur。
老板景天感到非常荣幸，忍痛拿出了李三思家祖传的铁观音来请 Arthur，作为答谢，Arthur 掏出一支狼毫，在地上画了个惟妙惟肖的菱形，唰的一闪，地上出现了一枚等大的钻石。
看着呆若木鸡口水直流的景天，Arthur 微微一笑“景老板请吧……”

输入：

一组测试数据。
第一行为一个整数 $n(0 \leq n \leq 20)$ ，表示钻石的数目。
接下来 n 行，每一行一个整数，为每块钻石的尺寸。

输出：

对于每个尺寸，输出对应的钻石，用*填充，具体形式参见输出样例。

输入样例：

```
3
2
3
5
```

输出样例：

```
*
***
*
*
```

```

***
*****
***
*
*
***
*****
*****
*****
*****
*****
***
*

```

对于这种输出的题目。

我只想说是让暴风雨来得更猛烈些吧!!

代码:

```

#include<iostream>

#include <stdio.h>
#include <cmath>
#include <string>
#define debug(x) cout <<#x<<"="<<x<<endl;//头文件多了不少。请无视。。
using namespace std;
int main()
{
    int n;
    int a,hs,hs1;
    cin>>n;
    while (n-->0)
    {
        cin>>a;
        for (hs=1;hs<=2*a-1;hs++)//这里设置了 hs 就是行数这个量
        {
            if (hs>a)

```

```

        hsl=2*a-hs;
        /*这个地方很重要，就是这种题后半部分是和前面一样的，就直接通过改变 hs
        这个量实现。不用在想办法,所以设置了 hsl*/
        else
        hsl=hs;
        for (int i=1;i<=a-hsl;i++)
        cout <<" ";
        for (int i=1;i<=2*hsl-1;i++)
        cout <<"*";
        cout <<endl;
    }
}
}

```

B 逃出迷宫

Problem Description

小蛮和龙幽两人在江湖上行侠仗义，锄强扶弱。老百姓非常感谢他们，但他们同时也结下了不少仇家。很明显，这些仇家都不是什么见得了人的人物，干得全都是下流手段。某天他们向小蛮和龙幽下了战帖，说要来一场光明正大的决斗，如果他们赢了，以后小蛮和龙幽就不得再干涉他们的事。龙幽觉得其中有诈，刚想拒绝。可是任性的小蛮却一口答应了下来，龙幽（一 一+）.....

等他们到了约定的地方却发现对方没有按时到，龙幽敏锐的察觉到事情不对，拉着小蛮就要离开，可这时虚空中传来了声音：“既来之，则安之。何必这么急着走呢.....这个阵法是专门为你们留的，找不到破阵之法，你们就将永远被困在里面了，哈哈哈哈.....”小蛮大声斥责他们无赖，龙幽无奈的回答“他们本来就是无赖(一.一|||，我们还是想办法出去吧”说完开始研究这个阵法.....

经过一番研究，龙幽发现此阵很奇特，每一步都得走特定的步数，一步错则得从头开始。经过一番尝试，他发现前几步满足下面这个数列.....

第一项：1 第二项：2 第三项：5 第四项：26 第五项：677.....

现在请你帮助他们逃出这座迷阵吧.....

Input

输入包含多组数据，每组数据为一行，为一个整数 n ($1 \leq n \leq 100$)。

Output

对于每一个 n ，输出他们第 n 步需要走的步数。现在只要求你给出这个步数的十位数的值。

Sample Input

```
3
5
6
```

Sample Output

```
0
7
3
```

找规律神马的最喜欢了。。。

别忘记一边平方一边取模。。

代码：

```
#include<iostream>
#include <stdio.h>
#include <cmath>
#include <string>
#define debug(x) cout <<#x<<"="<<x<<endl;
using namespace std;
int main()
{
    int n, sum;
    while (cin>>n)
    {
        sum=0;
        for (int i=1;i<=n;i++)
            sum=(sum*sum+1)%100;//模 100。。。
        sum/=10;//取十位数
    }
}
```

```
        cout <<sum<<endl;
    }

}
```

C 召唤英灵

题目描述

第 X 次圣杯战争即将拉开，作为参与其中的魔术师之一，**Darling** 首先要做的，是绘制召唤法阵。已知绘制法阵需要若干种化学元素，原本这对于 **Darling** 并非什么难事，但是 **Darling** 近期有点忙，因此她来找你帮忙。

你的任务很简单，给你所有原子序数，请你将 **Darling** 需要的元素名称写下来并交给化学药品店。

输入

一组测试数据。

第一行包含一个数 n ($1 \leq n$)，表示 **Darling** 绘制法阵需要的元素个数。

第二行包含 n 个整数，用空格隔开，为每种元素对应的原子序数。

输出

输出一行，为所需的全部元素名称（化学符号），元素与元素之间用一个空格隔开。（如果元素不在前四周期，则该元素名称为 **Usl**）。

具体参见输出样例

输入样例

```
3
1 17 50
```

输出样例

H Cl Usl

对于这种题目我还能说什么。。。化学渣渣伤不起啊。。。这是逼着我们复习化学知识啊。。。（助教用心良苦。。）

代码：

```
#include<iostream>

#include <stdio.h>
#include <cmath>
#include <string>
#define debug(x) cout <<#x<<"="<<x<<endl;
using namespace std;
int main()
{
    int n,a;
    cin>>n;
    string hx[37]={"H","He",
                  "Li","Be","B","C","N","O","F","Ne",
                  "Na","Mg","Al","Si","P","S","Cl","Ar",
                  "K","Ca","Sc","Ti","V","Cr","Mn","Fe",
                  "Co","Ni","Cu","Zn","Ga","Ge","As","Se","Br","Kr"}; /*我(sang)
不(xin)说(bing)话(kuang)*/
    while (n--)
    {
        cin>>a;
        if (a>36)
            cout <<"Usl ";
        else
        {
            cout <<hx[a-1]<<" ";
        }
    }
    /*注意减一。数组都是从 0 开始计数的*/
```

```
}  
  
}  
  
}
```

D 盗墓笔记之禁婆的考验

题目描述

推开那扇坑爹的石门，忽然一只冷冰冰的手搭在了 Thor 的肩上。回头，一张脸惊得他毛骨悚然——居然是——Arthur? !

长着一张 Atrhur 的脸的禁婆露出一个妩媚的微笑。

想过我这一关？先来玩个小游戏吧。很简答，给你一个十进制的数字 n ，你能求出它在 m 进制下的表示里数字 k 的个数么？

于是加法与求模都不会做的 Thor 又逗比了，怎么办？看你喽~

输入

多组测试数据。

每组数据为一行，包含三个整数 n, m, k ($2 \leq m \leq 8, 0 \leq k \leq 9$)。保证 n 在 `int` 范围内。

输出

对于每组数据，输出一行，包含一个整数，为 k 的个数。

输入样例

```
9 2 1  
8 8 1
```

输出样例

2
1

这个题目还是不错的。旨在让我们复习十进制化为别的进制的方法吧。。

代码：

```
#include<iostream>
#include <stdio.h>
#include <cmath>
#include <string>
#define debug(x) cout <<#x<<"="<<x<<endl;
using namespace std;
int main()
{
    int n,m,k,t;
    int js;//用来计数。。
    while (cin>>n>>m>>k)
    {
        js=0;//数据初始化的位置需要特别注意！！
        while (n>0)//注意条件的选择。
        {
            t=n%m;//就是一遍一遍的运算。。
            n=n/m;/*注意这个和上面的顺序不要颠倒。（好吧，
这是废话）*/
            if(t==k)
                js++;
        }
        cout <<js<<endl;
    }
}
```

E 盗墓笔记之怒海潜沙

题目描述

西沙水下，一阵暗流，Thor 与天真一行人被冲散了。

幸运的是，他被冲到了一扇门前。

不幸的是，刚刚到达这扇门前，门就合上了。上面的机关被重置为初始状态。

经过仔细的观察，Thor 发现，机关为一个圆盘，分为内外两圈。内圈均匀刻着 a 个小格，外圈均匀刻着 b 个小格，每个刻度有一个独一无二的图腾与之对应。

他还发现，每经过一分钟，内外两圈会同时顺时针旋转一个单位。从身边墙壁上的铭文，他得知，当轮盘再次回到初始状态时，门会重新打开。他想知道从门被合上到再次打开的这一个轮回需要多少时间，这个任务就交给你了。

输入

多组测试数据。

对于每组测试数据，输入两个数 a, b ($1 \leq a, b \leq 30000$)，用空格隔开，具体含义见题目描述。

输出

对于每组数据，输出一行，包含一个整数，表示一个轮回需要的时间。

输入样例

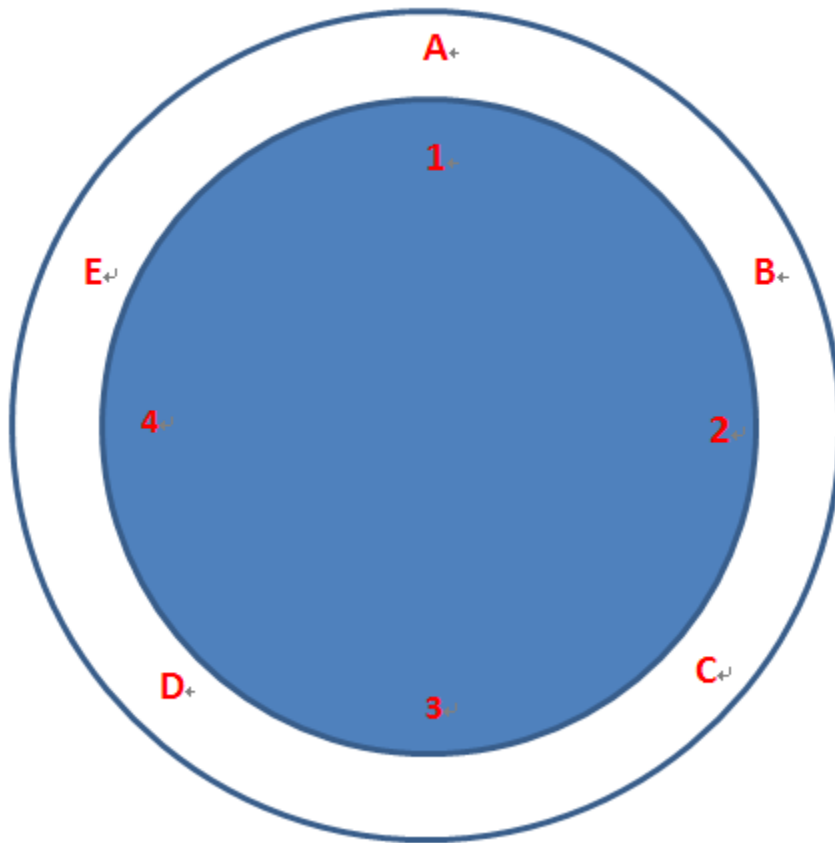
```
3 4
4 8
12 16
```

输出样例

```
12
8
48
```

HINT

机关？大概.....就是这个样子吧.....OTL



大家都能看出来，就是求最小公倍数。方法是先求最大公约数，再用两数相乘除掉最大公约数。

代码：

```
#include<iostream>
#include <stdio.h>
#include <cmath>
#include <string>
#define debug(x) cout <<#x<<"="<<x<<endl;
using namespace std;
int main()
```

```

{
    int a, b, a1, b1;
    int tem, ten; //两个暂时量。
    while (cin >> a >> b)
    {
        tem = 2;
        a1 = a;
        b1 = b;
        if (a < b)
        {
            tem = a;
            a = b;
            b = tem;
        } //确保 a > b。
        while (tem != 1 && tem != 0) /*最后时的两种情况，剩下 1 是最大
公约数就是 1. 剩下 0 的话最大公约数就是前一个被除的量。*/
        {
            tem = a % b;
            a = b;
            b = tem;
        } //辗转相除法
        if (tem == 1) //讨论一下那两种情况。
            cout << a1 * b1 << endl;
        else
            cout << a1 * b1 / a << endl;
    }
}

```

F 盗墓笔记之逃亡

题目描述

电闪雷鸣，警笛大作——不明人士的举报使得乘着列车去往长白山的 **Thor** 与天真不得不搞了一次夜半逃亡之旅。给你一上帝视角，将他们跑路的路线与警方的追查路线简化为一平面直角坐标系上的两条直线，由于警方人手的优越性，只要 **Thor** 与天真有可能出现在对方的路线上，就会被包围逮捕。

他们只能找一条完全与警方没有交集的路线，才可以安全逃离。现请给你这两条路线的一般表示（ $Ax+By+C=0$ ），请你判断他们是否会遭到逮捕。

输入

多组测试数据。

每组测试数据分为两行。第一行包含三个整数 A_1, B_1, C_1 ，用空格隔开，为 Thor 和天真的逃跑路线。

第二行包含三个整数 A_2, B_2, C_2 ，用空格隔开，为警方的追捕路线。

保证输入数据可以构成直线。

输出

对于每组数据，输出一行。如果 Thor 与天真会被逮捕，输出“Dead End”。如果他们安全逃离，则输出“Safe and Sound”

输入样例

```
1 1 1
1 1 2
1 1 1
1 2 1
```

输出样例

```
Safe and Sound
Dead End
```

平行线没香蕉。。。=。= 不开森。。

代码：

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

```

#include <cmath>
#include <string>
#define debug(x) cout <<#x<<"="<<x<<endl;
using namespace std;
int main()
{
    int a1,b1,c1,a2,b2,c2;
    while (cin>>a1>>b1>>c1>>a2>>b2>>c2)
    {
        if (a1*b2==a2*b1&&a1*c2!=a2*c1)/*平行的充要条件。什么，
你没学。。这个这个，初中的数学老师是谁教的。。*/
            cout <<"Safe and Sound\n";
        else
            cout <<"Dead End\n";
    }
}

```

G 李逍遥的仙剑客栈

Description

有一天天臧散人 **Arthur** 到了渝州东南的仙剑客栈，发现李逍遥这小子粗心大意在上酒的时候，有的桌上多上了酒，有的桌上没上酒。唉，谁让 **Arthur** 心软呢，不忍心看李逍遥被他婶婶骂，决定帮他——收拾一排桌子。

给你一个数 n 表示有多少个桌子，接下来给你一段整数序列表示每个桌上需要的酒量（假设这些桌子在一条直线上，且每个桌子之间的距离都是 1），正数表示多放了几瓶酒，负数表示应该放多少瓶酒。请你帮 **Arthur** 算一下他提着酒走的最短路程是多少。对了，**Arthur** 体力太渣，一次只能拿一瓶酒。

Input

第一行一个数 T 表示有 T 组数据。

接下来 T 组数据，每组数据有 2 行。

第一行一个数 $n(1 \leq n \leq 1000)$ ，表示桌子数量，

接下来第二行有一段数列 $A_i(|A_i| \leq 1000)$ ，表示每个桌子上应该放的酒。保证数列总和为 0。

Output

对于每组测试数据，输出一个数，表示 Arthur 拿着酒走的最少总路程。

Sample

Input:

```
2
3
-1 2 -1
6
-1 -1 -1 3 -1 1
```

Output:

```
2
7
```

Hint

其实就是每瓶酒走的路程之和。

我还是做的麻烦了。。

这个题真锻炼思维啊。。

代码:

```
#include<iostream>

#include <stdio.h>
#include <cmath>
#include <string>
#define debug(x) cout <<#x<<"="<<x<<endl;
using namespace std;
```

```

int main()
{
    int t, n, tem, az, bz, sum, js;
    int a[1000];
    cin>>t;
    while (t--)
    {
        cin>>n;
        sum=0;//用来计数的。
        for (int i=0;i<n;i++)
            cin>>a[i];
        for (int i=0;i<n-1;i++)
        {
            for (int j=i+1;j<n;j++)
            //两个 for 连用。用来对每个 ai 和后面的比一次。
            {
                if (a[i]*a[j]<0)//找到相反的那个桌子
                {
                    a[j]+=a[i];
                    a[i]=0;
                    //把酒全部放下。（或者拿起来）。不用管是不是正好
                    sum+=abs(a[i])*(j-i);//计数
                    break;
                }
            }
        }
        cout <<sum<<endl;
    }
}

```

这个题没必要找到那个相反的桌子，完全可以一个桌子一个桌子的前进。

H jhljx 学函数

Problem Description

jhljx 听说大家学了函数，决定考察大家的基本功。

给你两个数 **a** 和 **b**，请用函数来实现交换这两个数，使得 **a** 的值为 **b**, **b** 的值为 **a**。

本题必须用函数来完成。

不要在函数中先输出 **b,再输出 **a**。保证 **a** 的值是 **b**，**b** 的值是 **a**。请不要水过。**

Input

输入多组数据。

每组数据一行，为两个数 **a** 和 **b**。(**a** 和 **b** 在 `int` 范围内)

Output

输出进行交换后 **a** 和 **b** 的值。

Sample Input

1 2

Sample Output

2 1

Hint

函数声明的方法：

方法 1

```
int fuc(int);
int main()
{

}

int fuc(int a)
{

}
```

方法 2

```
int fuc(int a)
{

}

int main()
{

}
```

关于函数的值传递和引用传递

值传递

值传递是将数值传递给一个函数，但是函数中得到的数值只是原数值的一个副本。函数中对它进行操作，不会改变 `main` 函数中传递进来的那个参数的实际值。

```
void fuc(int m,int n)
{
    m++;n++;
    cout<<m<<" "<<n<<endl;
}

int main()
{
    int a,b;
    cin>>a>>b;
    fuc(a,b);
    cout<<a<<" "<<b<<endl;
}
```

如果输入 `2 3`，输出第一行为 `3 4`，第二行是 `2 3`。第一行的 `3 4`，是在函数中进行运算的结果。而第二行的 `2 3`，是原来 `a,b` 的值，说明调用 `fuc` 函数进行值传递时，没有改变原有 `a` 和 `b` 的值。

引用传递

引用传递是将数值传递给一个函数，函数中对数值进行操作会改变原来的值。

```
void fuc(int &m,int &n)
```

```
{
    m++;n++;
    cout<<m<<" "<<n<<endl;
}
int main()
{
    int a,b;
    cin>>a>>b;
    fuc(a,b);
    cout<<a<<" "<<b<<endl;
}
```

如果输入 2 3，就会输出两行 3 4。这说明在函数中 a,b 原来的值就已经改变了。

这个题目助教用心良苦啊。希望大家不要简单水过去。尽量把助教提供的方法都试试。。

代码 1:

```
#include<iostream>

#include <stdio.h>
#include <cmath>
#include <string>
#define debug(x) cout <<#x<<"="<<x<<endl;
using namespace std;
void fs(int a,int b )
{
    int c=a;
    a=b;
    b=c;
    cout <<a<<" "<<b<<endl;//直接用函数输出了。。
}
int main()
{
```

```

        int a,b;
        while (cin>>a>>b)
        {
            fs(a,b);
        }
    }
}

```

代码 2:

```

#include<iostream>

#include <stdio.h>
#include <cmath>
#include <string>
#define debug(x) cout <<#x<<"="<<x<<endl;
using namespace std;
void fs(int &a,int &b )//这里用引用传递。
{
    int c=a;
    a=b;
    b=c;
}
int main()
{
    int a,b;
    while (cin>>a>>b)
    {
        fs(a,b);
        cout <<a<< "    " <<b<<endl;//在输出。。
    }

}

```


I Ryan's Derivatives

Problem Description

Ryan 最近在阮姐的课上学习了导数==

有一天阮姐大笔一挥写下了一个多项式：

$$f(x) = a_0x^{b_0} + a_1x^{b_1} + \dots + a_{n-1}x^{b_{n-1}} + a_nx^{b_n}$$

然后又大笔一挥写下了另一个东西：

$$\frac{d^m f(x)}{dx^m}$$

然后说：求出它的表达式。这个当作课后作业==

数学很烂的 Ryan 怎么能禁受得住阮姐的轰炸，只好让你来帮助他完成阮姐的课后作业。

Input

多组测试数据。

每组测试数据包含 $n+1$ 行。

第一行为两个整数 n, m

接下来 n 行，每行两个整数， a_i 和 b_i

对于每组数据，所有 b_i 非负且均不相同

Output

对于每组数据，输出一行，为 $f(x)$ 的 m 次导数的表达式。

所有系数和指数应符合多项式书写的基本原则。

如：5 倍的 x 四次方 表示为 $5x^4$

1 倍的 x 不应表示为 $1x^1$ 而是 x

$5x^3+0x^2+(-x)$ 应表示为 $5x^3-x$

输出结果各项顺序与输入一致

Sample Input

```
4 2
1 4
2 3
3 2
4 1
3 1
1 5
-1 1
1 0
```

Sample Output

```
12x^2+12x+6
5x^4-1
```

我觉得我顺便把公告放这吧。。

H 题坑爹数据，在添加“+”的时候会出现错误。经过实际测试，最不容易考虑到的数据应该是这样的：

```
4 1
1 0
0 0
5 2
3 1
```

此外，需要说明的是本题比较人性的是 m 的值应该是正的，不会出现 0 次导数或者是负数次导数的情况，负数次导数就是积分，如果有这种数据，呵呵呵呵呵呵。首先你们会被积分和求导玩死。因此，忽略高等数学中的这些严格概念。只考虑求导的情形。

还有要注意这道题 b 是非负数。所以不会出现负数幂的情形。也就不需要考虑 x 的 -1 次方这样的情形。只用对 $b==0$ ， $b==1$ ， $b>1$ 进行分类即可。

([13211026](#) 2014-11-05 10:51:46)

大家可以参考一下。主要对分类讨论的考察比较严格。

还有如果所有求导结果都为 0 的话要进行特殊判断。直接输出 0。

代码：

```
#include<iostream>

#include <stdio.h>
#include <cmath>
#include <string>
#define debug(x) cout <<#x<<"="<<x<<endl;
using namespace std;

int main()
{
    int m,n,js;
    while (cin>>n>>m)
    {
        long long a[999]={0},b[999]={0},c[999]={0},d[999]={0};
        //我这为了避免麻烦直接弄了 4 个数组。分表代表系数、幂。求导后的系数和幂
        js=0; /*计数的初始化。。为了+号的出现和 0。。就是说+号只
        能在数字后面。0 就是没有任一个数输出时。*/
        for (int i=0;i<n;i++)
            cin>>a[i]>>b[i];
        for (int i=0;i<n;i++)
        {
            if (b[i]<m)//第一种。就是不够求导的，就是 0 了
                c[i]=0;
            else
            {
                c[i]=a[i];
                for (int j=0;j<m;j++)
                {
                    c[i]*=b[i];
                    b[i]--;
                    d[i]=b[i];
                } //求 m 次导的结果。。
            }
        }

        } //到这里就求导完了。。但是此题难在输出。。
        for (int i=0;i<n;i++)
        {
```

```

        if (c[i]!=0)//如果系数为 0 不输出
        {
            if(d[i]==1)//次数为 1 的。
            cout <<c[i]<<"x";
            else if(d[i]==0)//次数为 0 的
            cout <<c[i];
            else
            cout <<c[i]<<"x^"<<d[i];//正常的情况
            js++;//每输出一项就加一。
        }
        if(c[i+1]>0&&js!=0)
//如果下一项是正的，且不是第一项。如果是负的不输出了，有符号代替减号
        cout <<"+";
        if(i==n-1&&js!=0)//最后一项时换行。
        cout <<endl;
    }
    if (js==0)//没有项输出时就输出 0
    cout <<0<<endl;
}
}

```

此题就是要考虑的特别多。。特别多。。这才是真的桑兵。

J RecTangles! RecTangles! RecTangles!

题目描述

Arthur 这次也准备走一次简单路线，于是题面很简单。

给你平面直角坐标系下两个矩形中位于对角线上两个顶点的坐标，请判断这两个矩形是否会相互重叠。
矩形的两边与坐标轴平行，也就是说不会出现斜置的矩形。

输入

多组测试数据，每组数据为两行。

对于每组数据，第一行为四个整数，表示第一个矩形的两个对角顶点的坐标 (x1,y1) 与(x2,y2)。

第二行为四个整数，表示第二个矩形的两个对角顶点的坐标 (x3,y3) 与(x4,y4)。

(0≤x1,y1,x2,y2,x3,y3,x4,y4≤100)

输出

对于每组数据，如果相交，则输出“YES”，否则输出“NO”

输入样例

```
0 0 2 2
1 1 3 3
0 0 0 1
1 1 1 2
```

输出样例

```
YES
NO
```

HINT

重叠一个顶点或者一条边也算重叠
请考虑矩形合理性

这个题只有一个地方比较坑，就是矩形不合理时直接输出 NO。。当然中间的判断不太容易。。我的思路比较奇葩。就是先确定一个矩形。如果另一个矩形的顶点在这个内部，肯定是相交。然后再换过来。。（一开始没想到还得换过来再试一次。）

代码：

```
#include<iostream>

#include <stdio.h>
#include <cmath>
#include <string>
#define debug(x) cout <<#x<<"="<<x<<endl;
```

```

using namespace std;
void pp(int &a,int &b)
//这是第一个小函数。就是把两个数按从小到大排一下。排排。额。
//注意是引用传递的。这个函数用来确定大小关系。纯属懒。。
{
    int tem;
    if (a>b)
    {
        tem=a;
        a=b;
        b=tem;
    }
}
void hh(int &a,int &b)
//这是把两个数换一下。换换，，就是用来做第二次检验时用的
{
    int tem;
    tem=a;
    a=b;
    b=tem;
}
int pd(int x1,int x2,int x3)
{
//这是判定，就是判断一下第3个数是不是在前两个数之间
    if (x3>=x1&&x3<=x2)
        return 1;
    else
        return 0;
}
int main()
{
    int x1,x2,x3,x4,y1,y2,y3,y4;
    int tem,px,py;//临时数
    while (cin>>x1>>y1>>x2>>y2>>x3>>y3>>x4>>y4)
    {
        if (x1==x2||y1==y2||x3==x4||y3==y4)
            cout <<"NO\n";//就是矩形不存在时的情况。
        else
        {
            px=0;
            py=0;//初始化
            for(int i=1;i<=2;i++)//重复两次。

```

```

        {
            pp(x1, x2);
            pp(y1, y2); //确定一下大小关系
            if (pd(x1, x2, x3)==1 || pd(x1, x2, x4)==1)
                px=1;
            /*如果 x3 在 x1, x2 之间或者 x4 在 x1, x2 之间。就是某个横坐标满足在矩形内
            因为矩形 4 个点的坐标是 x3, x4 和 y3, y4 的 2*2 组合。所以先考虑横坐标有一个符合的。*/

            if (pd(y1, y2, y3)==1 || pd(y1, y2, y4)==1)
                py=1; //纵坐标。。
            hh(x1, x3);
            hh(x2, x4);
            hh(y1, y3);
            hh(y2, y4);

            //把坐标换一下。用来做第 2 次检验。
        }
        if(px==1&&py==1)
            //两次检验，只要有一次满足横坐标和纵坐标在范围内就是相交了。
            cout <<"YES\n";
        else
            cout <<"NO\n";
    }
}
}

```

最后一个还没做完。。好吧我承认是太难了，所以懒得思考了。。