

首先感（tu）谢（cao）下两位助教，真是尽（sang）职（xin）尽（bing）责（kuang），在英语等科目的考试周和期中高数考试前发布练习赛，而且还是晚上 11.30 公布题目，辛苦了！

## A. 双层汉诺塔

### 题目描述

#### Problem Description

---

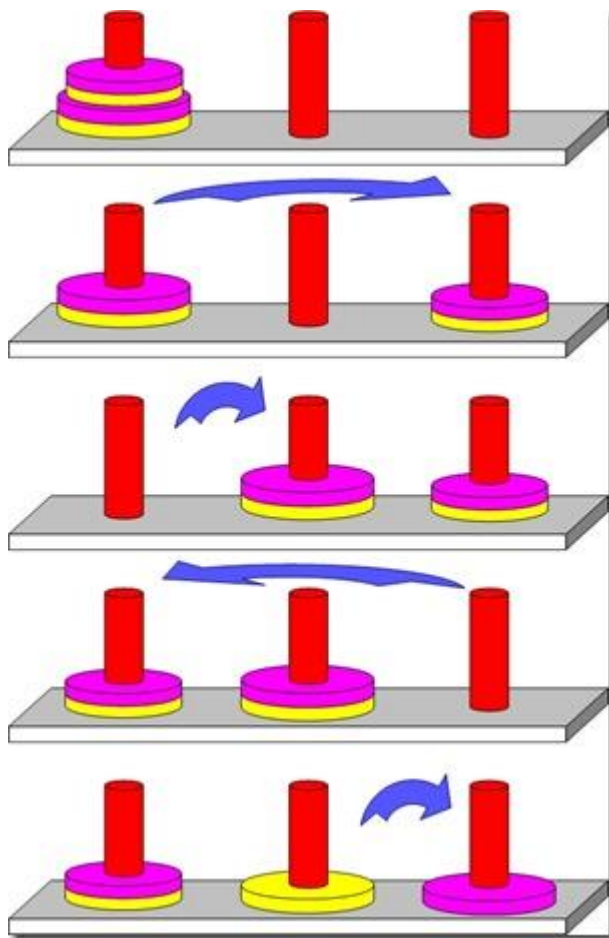
想必，前面的汉诺塔对于大家来说，just a piece of cake!

我们来玩一个更高级的汉诺塔游戏——双层汉诺塔。双层汉诺塔游戏中，相同大小的盘都有颜色不同的两片，按照次序放在最左边的金刚柱上。

游戏的目的是把不同颜色的盘分别按照上小下大的顺序放在中间和右边的两个柱子上。在移动过程中，在移动过程中，

依旧是遵守大盘必须在小盘之下，而颜色顺序无限制。

下面给出的四个盘子的例子，当然，这个图片是不完整的，还需要把左边粉红色的盘子移到最右边的柱子上，把黄色的盘子移动到中间的柱子上。



## Input

---

第一行输入一个整数  $t$  ( $t < 10$ )，表示测试的组数。

接下来的  $t$  行，每行输入一个正偶数  $n$  ( $0 < n \leq 20$ )，表示盘子的个数。

## Output

---

对于每组测试数据，首先输出一行 **Case #X:**，表示第  $X$  组测试数据。

接下来，每行都输出移动盘子的步骤，对于每次移动，以 "Move from X to Y" 表示，表示需要将  $X$  柱子上最顶层的盘子移动到  $Y$  柱子上，其中  $X, Y$  是 A,B,C 中的一个，当然， $X$  和  $Y$  不可能相同

## Sample Input

---

```
2
2
4
```

## Sample Output

---

```
Case #1:  
Move from A to B  
Move from A to C  
Case #2:  
Move from A to C  
Move from A to C  
Move from A to B  
Move from A to B  
Move from C to A  
Move from C to A  
Move from B to C  
Move from A to B  
Move from A to C
```

## 解题思路

双层汉诺塔是对汉诺塔的变形，套用一层汉诺塔即可很快做出，难度不大。没做出来的同学可以对比一下一层汉诺塔和二层汉诺塔的区别。寻找规律。

## 参考代码

```
#include <iostream>  
using namespace std;  
void h(int,char,char,char);  
void h_d(int,char,char,char);  
int main()  
{
```

```

int t, n, i=1;
cin >> t;
char x='A', y='B', z='C';
while(t--)
{
    cin >> n;
    cout << "Case #" << i << ":" << endl;
    h_d(n,x,y,z);
    i++;
}
}
void h(int n, char x, char y, char z)
{
    if(n==1)
    {
        cout << "Move from " << x << " to " << z << endl;
        cout << "Move from " << x << " to " << z << endl;
    }
    else
    {
        h(n-1,x,z,y);
        cout << "Move from " << x << " to " << z << endl;
        cout << "Move from " << x << " to " << z << endl;
        h(n-1,y,x,z);
    }
}
void h_d(int n, char x, char y, char z)
{
    for(int i=n/2; i>1; i--)
    {
        h(i-1,x,y,z);
        cout << "Move from " << x << " to " << y << endl;
        cout << "Move from " << x << " to " << y << endl;
        h(i-1,z,y,x);
        cout << "Move from " << y << " to " << z << endl;
    }
    cout << "Move from " << x << " to " << y << endl;
    cout << "Move from " << x << " to " << z << endl;
}

```

## B. Two Bags of Chocolates

### 题目描述

#### Problem Description

Darkness Wong 有两袋巧克力，已知第一袋里有  $x(x \geq 1)$  块，第二袋里有  $(y \geq 1)$  块。

很多周以前我们就知道，Darkness 是一个很粗心的人，毫不意外地，他把第一袋巧克力丢了。

Darkness 还记得两袋巧克力总块数  $(x+y)$  的一些特征。首先， $(x+y)$  不大于一个特定的正整数  $N$ ；其次， $(x+y)$  可以被正整数  $k$  整除。

帮助 Darkness 计算出第一袋中可能的巧克力块数  $x$ 。请升序输出所有可能的结果。

#### Input

多组测试数据。

每行测试数据只有一行，三个正整数  $y, k, n(1 \leq y, k, n \leq 10^9; (n/k) \leq 10^5)$ 。

#### Output

对于每组数据，输出一行，升序输出  $x$  所有可能的结果。

如果不存在满足条件的  $x$ ，输出 -1。

#### Sample Input

```
10 1 10
10 6 40
```

#### Sample Output

```
-1
2 8 14 20 26
```

## Source

An adaption of Codeforces Round #148(Div.2) Problem-A

## 解题思路

数学题目，判断满足 $(x+y)$ 不大于一个特定的正整数  $N$  和 $(x+y)$ 可以被正整数  $k$  整除两个条件的  $x$  的个数。对于两个条件进行判断即可

## 参考代码

```
#include <iostream>

using namespace std;
```

```

int main()
{
    long long x, y, n, k;
    while ( cin >> y >> k >> n )
    {
        int m = 0;
        if ( y >= n || k > n )
            cout << "-1" << endl;
        else
        {
            int i = 1;
            while ( k * i <= n )
            {
                if ( k * i > y )
                {
                    x = k * i - y;
                    cout << x << " ";
                    m = 1;
                }
                i++;
            }
            if ( m == 0 )
                cout << "-1" << endl;
            else
                cout << "\n";
        }
    }
}

```

## C. 圆有点挤

### 题目描述

#### 描述

---

gg 最近想给女友送两个精美的小礼品：两个底面半径分别为  $R_1$  和  $R_2$  的圆柱形宝石，并想装在一个盒子里送给女友。好不容易找到了一个长方体的盒子，其底面为  $A*B$  的矩形，他感觉好像宝石装不进去，但又不敢轻易塞进去试试。现请你帮他判断两个宝石能否放进盒子里(宝石只能竖直放置，且不能堆叠)。

#### 输入

---

输入的第一行是一个整数，为数据的组数  $t$  ( $t \leq 1000$ )。

每组数据占一行，包括 4 个数  $A$ ,  $B$ ,  $R_1$ ,  $R_2$ ，均为不超过  $10^4$  的正整数。

#### 输出

---

对于每组数据，若两个宝石能放进盒子中，则输出 YES，否则输出 NO。

#### 样例输入

---

```
2
10 10 1 1
10 10 4 4
```

#### 样例输出

---

```
YES
NO
```

#### 来源

---



## 解题思路

题目本质为判断两个半径分别为  $R_1$  和  $R_2$  的圆能否放进  $A*B$  的矩形中，  
显然最好的放法是一个紧靠左上角，另一个紧靠右下角。

因此只需判断数据能否满足以下两个条件：

<1> 两个圆都能放进矩形中，即： $2*R_1 \leq A$   
 $2*R_1 \leq B$   $2*R_2 \leq A$   $2*R_2 \leq B$ ；

<2> 采用紧靠对角放法后，两圆圆心的距离不小于半径之和；

即： $\text{sqrt}((A-R_1-R_2)^2 + (B-R_1-R_2)^2) \geq R_1 + R_2$

本题数据全为 int 型，为避免不必要的精度问题，可以将两边平方再进行比较，

即： $(A-R_1-R_2)^2 + (B-R_1-R_2)^2 \geq (R_1+R_2)^2$

## 参考代码

```
#include<iostream>
using namespace std;
int main()
{int n,A,B,R1,R2;
  while (cin>>n)
  {for(int i=1;i<=n;i++)
    {cin>>A>>B>>R1>>R2;
      if(( 2*R1<=A && 2*R1<=B && 2*R2<=A && 2*R2<=B )
        && ((A-R1-R2)*(A-R1-R2)+(B-R1-R2)*(B-R1-R2)>=(R1+R2)*(R1+R2) )) //重要判断
        cout << "YES"<<endl;
      else cout <<"NO"<<endl;
    }
  }
}
```

## D. 晴天小猪的妹妹

### 题目描述

#### Limit

Time Limit : 2 s

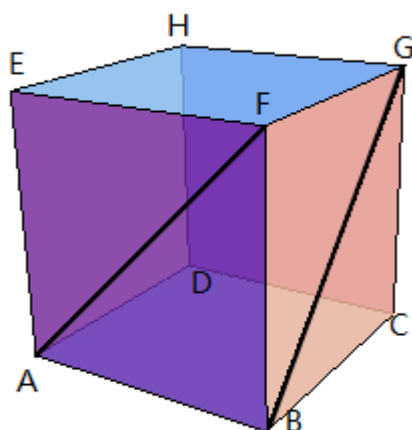
Memory Limit : 65535 KB

#### Description

晴天小猪是一个正儿八经的人。

但是晴天小猪也有妹子，还是个萝莉。

这天，小萝莉要高考了，他跑过来问哥哥一个题目：



长方体  $ABCD-EFGH$  中， $AD=x$ ， $DC=y$ ， $AE=z$ ，问直线  $AF$  和直线  $BG$  之间的距离是多少？

晴天小猪不会，他来求助于你。

## Input

第一行一个整数  $T$ ，为数据组数。

以下  $T$  组测试数据。

每组测试数据一行，为 3 个整数  $x,y,z$  ( $1 \leq x,y,z \leq 10000$ )

## Output

每组测试数据输出一行，表示两直线之间的距离。保留小数点后 5 位有效数字。

## Sample Input

```
1
1 1 1
```

## Sample Output

```
0.57735
```

## Source

GG

## 解题思路

求长方体中两条异面直线的距离，高考题目！（妈蛋我推了半天才会感觉高中白上了）推出公式为： $x*y*z/$

根号下( $x^2y^2+y^2z^2+z^2x^2$ ) ,之后再注意下保留五位小数即可。

## 参考代码

```
#include<iostream>
#include<cmath>
#include<iomanip>
using namespace std;

int main()
{ int T;
  double x,y,z,ans;
  cin>>T;
  for(int i=1;i<=T;i++)
  {
    cin>>x>>y>>z;
    ans=x*y*z/sqrt(x*x*y*y+y*y*z*z+z*z*x*x);
    cout<<setprecision(5)<<fixed<<ans<<endl;
  }
}
```

# E. 伤心的晴天小猪

## 题目描述

### Problem Description

---

上次上机后，好多同学居然责怪可爱的晴天小猪。笨笨的晴天小猪没有办法，遇到各种问题只能找聪明的你们求助。既然大家都不喜欢晴天小猪，伤心的他决定乘坐时光机去另一个时间点，摆脱这个伤心地。但是这个时光机太奇怪了，居然可以随意设定时间，10月50日都能设定，到达这个时间点估计就灰飞烟灭了。可是笨笨的晴天小猪不知道哪些时间是存在的，于是他只能再次向聪明的你们求助了。相信善良的你们不会故意让他灰飞烟灭的吧^\_^（画外音：就是要让他永远消失→\_→.....）

### Input

---

第一行是数据组数  $n$ ，接下来  $n$  组数据，每组数据一行按照年，月，日，小时，分钟，秒给定一个时间。（20: 00: 00 将以 20:0:0 给出，详见样例）

### Output

---

如果这个时间点合法（不考虑闰秒-\_-||）请输出 Good bye! 否则请输出 Please wait for a moment!

### Sample Input

---

```
2
2011 11 2 19 30 0
2011 2 29 1 20 35
```

### Sample Output

---

```
Good bye!
Please wait for a moment!
```

## 解题思路

本题和某次上机的时间格式的判断类似，但是要求和判断更多一些，加上了年月日，这就要求闰年和不同月里每月的日数不同等特殊情况的判断，所以多注意情况的判断，不要忘了哪月 30 天哪月 31 天和闰年这些情况分开来即可。

## 参考代码

```

#include <iostream>
using namespace std;
int y,m,d,h,mi,s,n;
void fun();
int main()
{
    while(cin>>n)
    {
        for(int i=1;i<=n;i++)
            fun();
    }
    return 0;
}

void fun()
{
    cin>>y>>m>>d>>h>>mi>>s;
    if(((y%4==0)&&(y%100!=0)) || (y%400==0)) //闰年
    {
        if((d>=1)&&(d<=29)&&(m==2)&&(h>=0)&&(h<=23)&&(mi>=0)&&(mi<=59)&&(s>=0)&&(s<=59))
            cout<<"Good bye!"<<endl;
        else
        if((d>=1)&&(d<=31)&&((m==1) || (m==3) || (m==5) || (m==7) || (m==8) || (m==10) || (m==12))&&(h>=0)&&(h<=23)&&(mi>=0)&&(mi<=59)&&(s>=0)&&(s<=59))
            cout<<"Good bye!"<<endl;
        else
        if((d>=1)&&(d<=30)&&((m==4) || (m==6) || (m==9) || (m==11))&&(h>=0)&&(h<=23)&&(mi>=0)&&(mi<=59)&&(s>=0)&&(s<=59))
            cout<<"Good bye!"<<endl;
        else cout<<"Please wait for a moment!"<<endl;
    }
    else
    if((d>=1)&&(d<=28)&&(m==2)&&(h>=0)&&(h<=23)&&(mi>=0)&&(mi<=59)&&(s>=0)&&(s<=59))
        cout<<"Good bye!"<<endl;
    else
    if((d>=1)&&(d<=31)&&((m==1) || (m==3) || (m==5) || (m==7) || (m==8) || (m==10) || (m==12))&&(h>=0)&&(h<=23)&&(mi>=0)&&(mi<=59)&&(s>=0)&&(s<=59))
        cout<<"Good bye!"<<endl;
    else
    if((d>=1)&&(d<=30)&&((m==4) || (m==6) || (m==9) || (m==11))&&(h>=0)&&(h<=23)&&(mi>=0)&&(mi<=59)&&(s>=0)&&(s<=59))
        cout<<"Good bye!"<<endl;
    else cout<<"Please wait for a moment!"<<endl;
}

```



# F. 晴天小猪的绕口令

## 题目描述

### Problem Description

---

第二个游戏是绕口令。规则是：主持人给出一串字符串，要求把这串字母简化。该串字符串全部为小写英文字母。

比如：aaabbbcc，则简化为 3a3b2c；zzzzeeeeea，则简化为 4z5e1a。依次类推。

最后一题了，帮帮晴天小猪哦。

注意：本题禁止使用数组和字符串。否则此题记 0 分。

### Input

---

第一行为一个整数  $n$ ，表示共有  $n$  组测试数据（ $1 \leq n \leq 100$ ）。每组测试数据有一行，该行第一个数为字符串长度  $t$ （ $t \leq 1,000,000$ ），然后为一行长度为  $t$  的字符串。

### Output

---

对于每组输入数据输出一行，即简化后的字符串。

### Sample Input

---

```
3
7 aaaaaaa
4 abcd
6 qwweee
```

### Sample Output

---

```
7a
1a1b1c1d
1q2w3e
```

## 解题思路

我只想问一句晴天小猪是哪个别 ( sang ) 出 ( xin ) 心 ( bing ) 裁 ( kuang ) 的学长，题目设计的真的很巧妙呢！妈蛋 XXXXXXXXXXXX ( 此处和谐一万字 ) 嗯，没有用字符串和数组，恩，没有用 cin cout 妈蛋为什么还是 MLE !!!

最后删了 iostream 竟然过了。。泪都下来了。。对于还不知道 iosream 的渣比考这种 BT 的题目真的好么。。题目思路比较简单，就是字符的判断，计数器，然后输出。得出的结论就是考察头文件。( xxxxx )

## 参考代码

**//#include<iostream> (不要问我为什么没有这个，有钱！任性！)**

```
#include<cstdio>
using namespace std;
int n,t,num;
char ch,s;
int main()
{
    scanf("%d",&n);
    for(int i=1; i<=n; i++)
    {
        scanf("%d",&t);
        getchar();
        num=1;
        s='+'; //先给后面要用的 s 赋一个它后面一定不会出现的值
        for(int j=1; j<=t+1; j++)
        {
            if (j<=t)
            {
```

```

scanf("%c",&ch); //输入 t 个 char
}
else ch='-'; //第 t+1 个用一个它一定不会出现的值赋给 ch，以便最后一次的输出
出
if (s == ch) num++; //第一次的时候 s 是'+', 不相等，num 即你的 sum 不加
else if(s != '+') //第一次的时候 s 是'+', 不输出，因为就一个字符，还不需要输出
{
    printf("%d%c",num,s);
    num = 1;
}
s = ch; //在这里才赋值正确的 s，前面都是'+',
}
printf("\n"); //代码比较奇葩比正常顺序晚一次输出
}
}

```