

第五次练习赛解题报告

By——14211103 田旺

A 题：双层汉诺塔

Problem Description

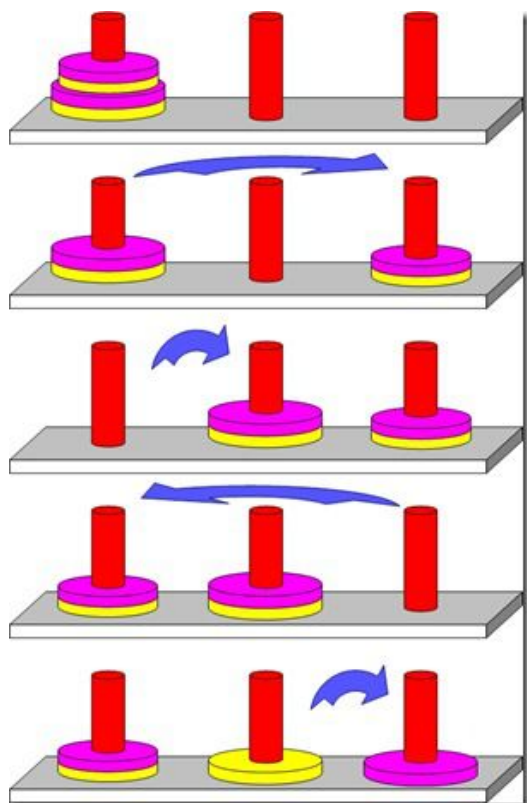
想必，前面的汉诺塔对于大家来说，just a piece of cake!

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，

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

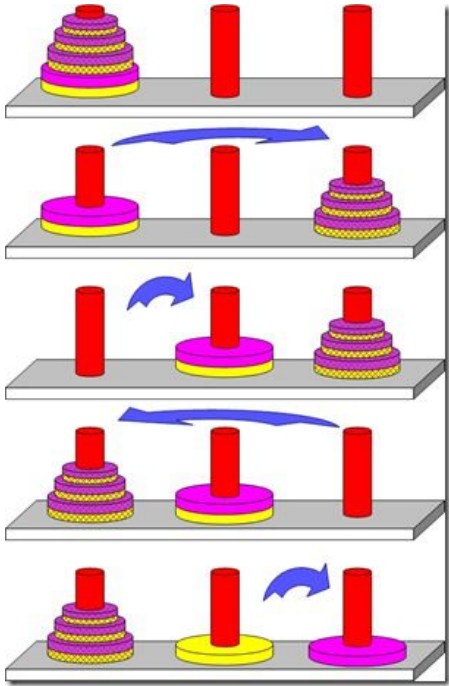
```
2
2
4
```

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：

当只有 4 个盘子时，就像如题所示的图移动就好。当盘子数大于 4 个时，将处最后两个盘子外的其它盘子看作一个整体，利用递归进行移动。如图所示：



参考代码：

```
#include<stdio.h>
#include<iostream>
using namespace std;
void hanoi(int,char,char,char);
void hanoi_double(int,char,char,char);
int main()
{
    int n;cin>>n;
    for(int i=1;i<=n;i++)
    {
        int x;cin>>x;char a='A',b='B',c='C';
        cout<<"Case #"<<i<<":"<<endl;
        hanoi_double(x,a,b,c);
    }
}
void hanoi(int n,char A,char B,char C)
{
    if (n==1)
    {
        printf("Move from %c to %c\n",A,C);
        printf("Move from %c to %c\n",A,C);
    }
    else
    {
        hanoi(n-1,A,C,B);
        printf("Move from %c to %c\n",A,C);
        printf("Move from %c to %c\n",A,C);
        hanoi(n-1,B,A,C);
    }
}
```

//这就是我们最熟悉的单层汉诺塔，但在这儿输出时，因为相同大小的盘子有两个，所以要输出相同的两次移动。

```
void hanoi_double(int n,char A,char B,char C)
{
    for (int i=n/2;i>1;i--)
    {
        hanoi(i-1,A,B,C);
        printf("Move from %c to %c\n",A,B);
        printf("Move from %c to %c\n",A,B);
        hanoi(i-1,C,B,A);
        printf("Move from %c to %c\n",B,C);
    }
    printf("Move from %c to %c\n",A,B);
    printf("Move from %c to %c\n",A,C);
}
```

}//将除最后的两个盘子之外的盘子看作一个整体，进行像单层汉诺塔那样的移动，再对剩下的两个盘子进行如题目所示的移动。

PS : 此 题 参 考 了 网 络 资 料

<http://blog.csdn.net/forfuture1978/article/details/477476>

4

B 题: Two Bags of Chocolates

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
```

解题思路：

此题应该是这一套练习赛中最简单的一道题，可以先令 $x=1$ ，再在满足 $x+y \leq n$ 的情况下，对 x 进行不断加 1 的操作，并在每加一步时就判断 $(x+y)$ 能否被 k 整除，如果是，就输出 x ，由于是对 x

由小到大开始加的，所以 x 就是按照从小到大的顺序输出的。另外可以通过，记录 x 的个数来判断是否存在满足条件的 x 。

参考代码：

```
#include<cstdio>
#include<iostream>
using namespace std;
int main()
{
    long long y,k,n;//感觉若是都定义 long long 类型，就保险多了。
    while(cin>>y>>k>>n)
    {
        if(y>n)//注意，一般都会只考虑  $x$  是否满足，但不要忘了  $y$  也有要求。
            cout<<-1<<endl;
        else
        {
            int sum=0;
            for(long long x=1;x+y<=n;x++)//在  $x+y$  不大于  $n$  的情况下，对  $x$  累加，挨个判断。
            {
                if((x+y)%k==0)//判断是否能被  $k$  整除。
                    cout<<x<<" ";
                else
                    sum=sum+1;//记录不满足的  $x$  的个数。
            }
            if(sum==n-y)//表示  $y$  到  $n$  之间的所有的  $x$  都不满足。
                cout<<-1;
            cout<<endl;
        }
    }
}
```

C 题：圆有点挤

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

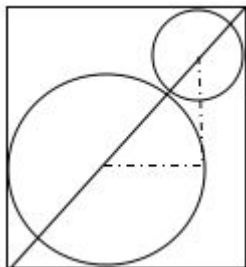
```
YES
NO
```

来源

第六届北航程序设计大赛现场决赛

解题思路：

要注意判断的是，圆怎么放才是满足最小矩形要求的，如下图：当圆按照矩形对角线的摆放时，是矩形能满足的最小条件。那么按照基本的三角函数知识计算判断就可以了。为了方便，可以通过转换，将 A 变为最大值，即长方形的长，并将 r1 记为两个圆中的较大的圆的半径。



参考代码：

```
#include<iostream>
using namespace std;
int main()
{
    int n;
    cin>>n;
    for(int i=1;i<=n;i++)
    {
        int a,b,r1,r2,s1,s2;
        cin>>a>>b>>r1>>r2;
        s1=a;
        if(b>=s1)
        {
            s1=b;
```

```

        b=a;
        a=s1;
    }//将 a 变为最大值。
    s2=r1;
    if(r2>=s2)
    {
        s2=r2;
        r2=r1;
        r1=s2;
    }//将 r1 变为最大值。
    if(b>=2*r1&&(b-r1-r2)*(b-r1-r2)+(a-r1-r2)*(a-r1-r2)>=(r1+r2)*(r1+r2))// 利用 直角 三
角形的斜边与两条直角边的关系判断。注意边长还必须满足至少能放进去一个圆。
        cout<<"YES"<<endl;
    else
        cout<<"NO"<<endl;
    }
}

```

D 题：晴天小猪的妹妹

Limit

Time Limit : 2 s

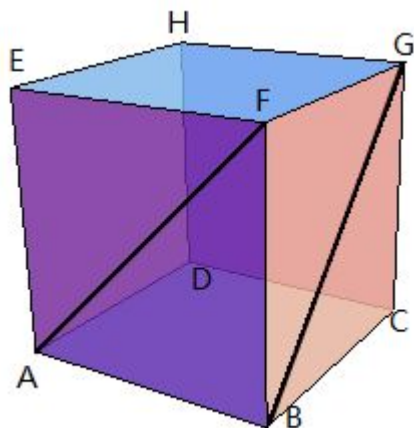
Memory Limit : 65535 KB

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：

瞬间感觉回到高三啊！有木有！！这就是高中时数学立体几何中，在空间直角坐标系中，利用向量方法计算空间中两条异面直线间的距离的题目。那么，就在草稿纸上算吧。写程序时计算最后关于 x, y, z 的表达式即可。

参考代码：

```
#include<iostream>
#include<cmath>
#include<iomanip>
using namespace std;
int main()
{
    int n;cin>>n;
    for(int i=1;i<=n;i++)
    {
```



```

        long double x,y,z;long double s;
        cin>>x>>y>>z;
        s=z/(sqrt((z*z)/(x*x)+(z*z)/(y*y)+1));//这就是最后的需要计算的表达式。
        cout<<fixed<<setprecision(5)<<s<<endl;
    }
}

```

E 题：伤心的晴天小猪

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!

```

参考代码：

```

#include<iostream>
using namespace std;
int main()
{
    int n;
    cin>>n;
    for(int i=1;i<=n;i++)
    {
        int y,m,d,h,f,s;
        cin>>y>>m>>d>>h>>f>>s;
        if(h>=0&&h<=23&&f>=0&&f<=59&&s>=0&&s<=59&&m>=1&&m<=12)//因为月，时，分，
秒是不论在什么情况下都是统一的，所以优先判断这个是否成立。

```

```

{
    if((y%4==0&& y%100!=0)||y%400==0)//这是判断闰年的方法。
    {
        if(m==1||m==3||m==5||m==7||m==8||m==10||m==12)
        {
            if(d>=1&&d<=31)
                cout<<"Good bye!"<<endl;
            else
                cout<<"Please wait for a moment!"<<endl;
        }//判断天数为 31 天的月份是否符合标准。
        if(m==2)
        {
            if(d>=1&&d<=29)
                cout<<"Good bye!"<<endl;
            else
                cout<<"Please wait for a moment!"<<endl;
        }//闰年的 2 月为 29 天，需单独判断。
        if(m==4||m==6||m==9||m==11)
        {
            if(d>=1&&d<=30)
                cout<<"Good bye!"<<endl;
            else
                cout<<"Please wait for a moment!"<<endl;
        }//判断天数为 30 天的月份是否符合标准。
    }
}
else//接下来就是判断平年的情况。唯一与闰年不同的就是 2 月的天数。
{
    if(m==1||m==3||m==5||m==7||m==8||m==10||m==12)
    {
        if(d>=1&&d<=31)
            cout<<"Good bye!"<<endl;
        else
            cout<<"Please wait for a moment!"<<endl;
    }
    if(m==2)
    {
        if(d>=1&&d<=28)
            cout<<"Good bye!"<<endl;
        else
            cout<<"Please wait for a moment!"<<endl;
    }//平年的 2 月为 28 天。
    if(m==4||m==6||m==9||m==11)
    {
        if(d>=1&&d<=30)
            cout<<"Good bye!"<<endl;
    }
}

```

```

        else
            cout<<"Please wait for a moment!"<<endl;
        }
    }
}
else//此处的 else 对应的是最开始判断月，时，分，秒是否符合要求。
    cout<<"Please wait for a moment!"<<endl;
}
}

```

F 题：晴天小猪的绕口令

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：

简单的判断思路为，对每一个输入的字符，与上一个进行比较，如果相同，就将一个计数变量加 1，如果不相同，就将新的输入数据作为新的比较标准，将下一个输入的字母与它比较，同时记得让计数变量归 1。还有更多细节会在代码注释中提到。

参考代码：

`#include<cstdio>` //注意，这道题卡头文件了，不能用 `#include<iostream>`。而且 `using namespace std`；也没准用。

`int main()` //为了避免超时，本题将全部采用 C 语言的输入输出。

```

{
    int n;scanf("%d",&n);
    for(int i=1;i<=n;i++)
    {
        int x;char a;char b;scanf("%d %c",&x,&a);int s=1;//x 为每一组的字符个数，先输入一个
        字符 a 作为最初的比较标准，注意红色部分的%d 和%c 之间的空格，这是为了规定输入格式，避免读入空格。s 为相同的字符的个数。
        if(x>=2)//因为后面会用到 x-1，所以将 x 大于 2 的情况分开执行。
        {
            for(int j=1;j<=x-1;j++)//因为前面已经输入了一个字符 a，所以接下来只需要再输入
            x-1 个字符。
            {
                scanf("%c",&b);
                if(a==b)
                    s++;//如果新输入的字符与前面的字符相同，就将计数变量加 1。
                else
                {
                    printf("%d%c",s,a);
                    a=b;s=1;
                }//当新输入的字符与前面的不不同时，就输出前面的相同字符的个数与字符。
                同时，将新的字符作为新的比较标准 a，同时，计数变量归为 1。
                if(j==x-1)
                {
                    printf("%d%c",s,b);
                }//由于前面的输出是输出前面的相同字符，所以最后部分的字符需单独输出。
            }
            printf("\n");
        }
        if(x==1)
            printf("%d%c\n",s,a);//当只有一个字符时，直接输出。
    }
}

```

