# 14级第四次上机解题报告

A临近的虚实

# Problem Description

KamuiKirito简洁明了。  
给你一个时间，判断这个时间是否合法。  
没有闰秒一类奇怪的东西  
不存在24:00:00,只有0:00:00  
注意前导0

# Input

多组测试数据。  
每组数据为一行字符串，格式见样例。

# Output

每组数据输出一行，合法输出Yes，不合法输出No。

# Sample Input

23:33:33  
5:05:05  
6:66:66

# Sample Output

Yes  
Yes  
No

# Hint

推荐用C语言格式的输入。  
当然不用也是可以AC的。

分析：

我们可以用C语言的格式输入；然后判断时、分、秒，是否合理就可以了；

A参考代码

#include<iostream>

#include<cstdio>

using namespace std;

int main()

{

int a,b,c;

while(scanf("%d:%d:%d",&a,&b,&c)!=EOF)

{

if(a>=0&&a<=23&&b>=0&&b<60&&c>=0&&c<60)

cout<<"Yes"<<endl;

else

cout<<"No"<<endl;

}

}

B正义的天平

# Problem Description

KamuiKirito简洁明了。  
给你一串字符，如果是字母则输出对应ASCII码，如果是数字则直接输出该数字

# Input

多组测试数据。  
每组数据有两行。  
第一行为n，为字符串长度。  
第二行为一串字符，长度为n，为英文字母或数字。

# Output

每组数据输出一行，为题目要求输出。

# Sample Input

3  
450  
8  
jhljx233

# Sample Output

450  
106104108106120233

分析：

只需要判断是字母还是数字就可以了；

B参考代码

#include<iostream>

using namespace std;

int main()

{

int n,c;

char a;

while(cin>>n)

{

while(n--)

{

cin>>a; if(a=='0'||a=='1'||a=='2'||a=='3'||a=='4'||a=='5'||a=='6'||a=='7'||a=='8'||a=='9')

cout<<a;

else

{c=a;

cout<<int(a);

}

}cout<<endl;

}

}

C恶魔的证明

# Problem Description

KamuiKirito简洁明了。  
给你一串字符，如果存在一个大写字母相邻的两个字母都是小写字母，则证明这串字符有恶魔存在。

# Input

多组测试数据。  
每组数据有两行。  
第一行为n，为字符串长度。  
第二行为一串字符，长度为n。

# Output

每组数据输出一行，如果存在恶魔输出Yes，否则输出No。

# Sample Input

5  
jhljx  
11  
KamuiKirito

# Sample Output

No  
Yes

分析:

C参考代码

#include<iostream>  
using namespace std;  
int main()  
{  
    int n;  
    while(cin>>n)  
    {  
        int m=n,o=0;  
        char a[n];  
        while(n--)  
        {  
            cin>>a[n];  
        }  
        for(int i=1;i<=m;i++)  
        {  
            if((a[i]=='A'||a[i]=='B'||a[i]=='C'||a[i]=='D'||a[i]=='E'||a[i]=='F'||a[i]=='G'||a[i]=='H'||a[i]=='I'||a[i]=='J'||a[i]=='K'||a[i]=='L'||a[i]=='M'||a[i]=='N'||a[i]=='O'||a[i]=='P'||a[i]=='Q'||a[i]=='R'||a[i]=='S'||a[i]=='T'||a[i]=='U'||a[i]=='V'||a[i]=='W'||a[i]=='X'||a[i]=='Y'||a[i]=='Z')&&(a[i-1]=='a'||a[i-1]=='b'||a[i-1]=='c'||a[i-1]=='d'||a[i-1]=='e'||a[i-1]=='f'||a[i-1]=='g'||a[i-1]=='h'||a[i-1]=='i'||a[i-1]=='j'||a[i-1]=='k'||a[i-1]=='l'||a[i-1]=='m'||a[i-1]=='n'||a[i-1]=='o'||a[i-1]=='p'||a[i-1]=='q'||a[i-1]=='r'||a[i-1]=='s'||a[i-1]=='t'||a[i-1]=='u'||a[i-1]=='v'||a[i-1]=='w'||a[i-1]=='x'||a[i-1]=='y'||a[i-1]=='z')&&(a[i+1]=='a'||a[i+1]=='b'||a[i+1]=='c'||a[i+1]=='d'||a[i+1]=='e'||a[i+1]=='f'||a[i+1]=='g'||a[i+1]=='h'||a[i+1]=='i'||a[i+1]=='j'||a[i+1]=='k'||a[i+1]=='l'||a[i+1]=='m'||a[i+1]=='n'||a[i+1]=='o'||a[i+1]=='p'||a[i+1]=='q'||a[i+1]=='r'||a[i+1]=='s'||a[i+1]=='t'||a[i+1]=='u'||a[i+1]=='v'||a[i+1]=='w'||a[i+1]=='x'||a[i+1]=='y'||a[i+1]=='z'))  
            {  
                o=o+1;  
            }  
        }  
        if(o>0)  
        cout<<"Yes"<<endl;  
        else  
        cout<<"No"<<endl;  
    }  
}

D约伯的救赎

# Problem Description

KamuiKirito的家乡下雪了。  
KamuiKirito发现n片雪花聚集在一起的形状跟n有关。  
n=1：  
#  
n=2：  
##  
n=3：  
##  
\*  
n=4：  
##  
\*\*  
n=5：  
##  
\*\*  
#

# Input

多组测试数据。每组测试数据只有一行为一个整数n（1<=n<=30）。

# Output

对于每组输入数据，输出雪花的样子。

# Sample Input

8

# Sample Output

##  
\*\*  
##  
\*\*

分析：

输入n时，我们可以求出总共输出有多少行（n+1）/2;

然后分行数为奇数（##），行数为偶数（\*\*）进行判断

最后还要判断n是否为奇数（为了判断最后一行是输出一个#还是\*）

D参考代码

#include<iostream>

using namespace std;

int main()

{

int n;

int a,b,x;

while(cin>>n)

{

a=n/2;

x=a;

b=n%2;

for(int i=1;i<=a;i++)

{

if(i%2==1)

cout<<"##"<<endl;

if(i%2==0)

cout<<"\*\*"<<endl;

}

if(x%2==1&&b==1)

cout<<"\*"<<endl;

if(x%2==0&&b==1)

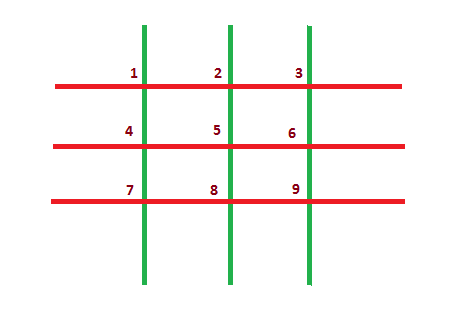
cout<<"#"<<endl;

}

}

E无法禁止的游戏

# Problem Description

KamuiKirito和jhljx玩游戏。  
一个由n条横线m条竖线构成的网格。  
例如一个3\*3的网格  
  
每次一方选择一个交点，将该点所在的横线和竖线删掉然后又另一方进行。  
进行时如果网格不存在交点则失败。  
jhljx大神先来。

# Input

多组测试数据  
每组测试数据一行，两个整数，为n,m。（1<=n,m<=100)

# Output

对于每组输入数据，输出一行，为获胜方。

# Sample Input

1 1  
2 2

# Sample Output

jhljx  
KamuiKirito

分析：

首先要明确游戏规则；

规则就是：两个人在一个网格里挑出点，并且删掉构成这个点的两条线；最后如果没有交点让另一个人挑那么，他就输掉游戏；

还有就是jhljx每次都先；

那么我们可以找到n,m的最小一个min；

然后将min--；同时int i=1;i++;最后判断i的奇偶来确定谁赢；

E参考代码

#include<iostream>  
using namespace std;  
void fuc(int a,int b)  
{  
  
}  
int main()  
{  
    int n,m,x=0;  
    while(cin>>n>>m)  
    {  
          int t=0;  
        if(n<m)  
        {  
            x=m;  
            m=n;  
            n=x;  
        }  
     while(m--)  
     {  
         t++;  
     }  
       //cout<<t<<endl;  
       if((t+1)%2==0)  
       cout<<"jhljx"<<endl;  
       if((t+1)%2!=0)  
       cout<<"KamuiKirito"<<endl;  
    }  
}

F掷石之人

# Problem Description

Creeper学妹每天都从五楼使用小型火焰烧楼下约会的XXX。  
jhljx有些忍不了了，于是Creeper学妹做出妥协。  
从今天（2014年11月14日）开始，每周五才使用奥义：火球术。剩下的日子由掷石子代替。  
给你一个日期，让你判断这天Creeper学妹是喷火还是掷石子。  


# Input

多组测试数据  
每组测试数据一行，为一个日期（保证合法）。

# Output

对于每组输入数据，输出一行，如果Creeper学妹喷火则输出Fire，否则输出Poor Creeper。

# Sample Input

2014-11-14  
2015-01-02  
2015-01-03

# Sample Output

Fire  
Fire  
Poor Creeper

# Hint

无限大な梦のあとの 何もない世の中じゃ

分析：

我们要做的就是判断这一天是不是星期五；可以用公式**w=d+2\*m+3\*(m+1)/5+y+y/4-y/100+y/400)% 7；**

**W=0 表示周一；w=1表示周二；w=2表示周三；w=3表示周四；w=4表示周五；w=5表示周六；w=6表示周日；**

**其中y表示年；m表示月份；d表示天；**

**计算时有特例当m为1或m为2时要将y减一同时m变为12+m;其他月份正常代入。**

**例如2014-01-02 那么就应该计算2013-13-02**

**又例如2014-02-02 那么就应该计算201-14-02**

F参考代码

#include<iostream>

#include<cstdio>

using namespace std;

int main()

{

int y,m,d,W;

while(scanf("%d-%d-%d",&y,&m,&d)!=EOF)

{

if(m==1||m==2)

{

y=y-1;

m=12+m;

W= (d+2\*m+3\*(m+1)/5+y+y/4-y/100+y/400)% 7 ;

if(W==4)

cout<<"Fire"<<endl;

if(W!=4)

cout<<"Poor Creeper"<<endl;

}

else if(m!=1&&m!=2)

{

W= (d+2\*m+3\*(m+1)/5+y+y/4-y/100+y/400)% 7 ;

if(W==4)

cout<<"Fire"<<endl;

if(W!=4)

cout<<"Poor Creeper"<<endl;

}

}

}