

E1 - Statement

A 摩卡当梦拓

题目描述

梦拓计划是北航的一项独特传统，旨在帮助刚入学的新生们快速适应大学生活，将北航精神与文化代代相传。

Moca 今年有幸继续担任了信息大类的一名梦拓，肩负起迎接新生的责任。作为新生们与北航的第一次正式接触，Moca 需要给梦拓小组的每位新生发送欢迎短信。短信的模板如下：

```
Greetings from BUAA, your Student ID is xxxxxxxx, Class Number is  
xxxxxx.
```

其中，Moca 需要将每位新生的学号和班级号替换到对应的位置。为了提高效率，Moca 希望能自动化这个流程。现在给定 n 名新生的学号和班级号，你能写一个程序帮 Moca 自动生成所有的新生短信吗？

输入

第一行一个正整数 n ，表示 Moca 的梦拓小组中新生的数量，有 $1 \leq n \leq 100$ 。

接下来 n 行，每行两个正整数 id 和 $class$ ，分别代表新生的学号和班级，其中学号是一个八位不以 0 开头的数字，班级号是一个六位不以 0 开头的数字。

输出

输出 n 行，分别代表 n 名新生对应的短信。

输入样例

```
2  
12345678 123456  
88888888 147258
```

输出样例

```
Greetings from BUAA, your Student ID is 12345678, Class Number is  
123456.  
Greetings from BUAA, your Student ID is 88888888, Class Number is  
147258.
```

Hint

基本输入和格式化输出，可以参考课件 P1 例 1-3

你可以参考下面的程序主体框架：

```
scanf("%d", &n);
for(int i = 0; i < n ;i++) {
    do something ...
}
```

这样，就能在读入一个整数 n 后，执行同样的一段代码 n 次

其次，注意是要输出 n 行，别忘了换行符

std

```
#include <stdio.h>

int main() {
    int n;
    scanf("%d", &n);

    for(int i = 1; i <= n ;i++) {
        int id, class;
        scanf("%d%d", &id, &class);
        printf("Greetings from BUAA, your Student ID is %d, Class
Number is %d.\n", id, class);
    }
    return 0;
}
```

B cyz 买文具

题目描述

cyz 同学作为班级里的学习委员，在老师的要求下去文具店买尽可能多的笔，文具店内一支笔的价格为 2 元 5 角 9 分钱，而班费有 x 元 y 角 z 分，cyz 同学想知道，他最多能买多少支笔？

输入

输入只有一行 3 个整数，分别为 x, y, z

输出

输出一行一个整数，表示 cyz 同学最多能买多少支笔

输入样例

```
12 6 8
```

输出样例

```
4
```

数据范围

对于全部数据，保证 $0 \leq x \leq 100, 0 \leq y \leq 9, 0 \leq z \leq 9$

std

```
#include<stdio.h>
int x,y,z;
int main()
{
    scanf("%d%d%d",&x,&y,&z);
    int money = x * 100 + y * 10 + z;
    printf("%d",money / 259);
    return 0;
}
```

C 摩卡与小学期

题目描述

Moca 刚刚结束了她的小学期。

在小学期中，4 到 6 个同学将组成一个小组，共同完成若干需求。现在 Moca 将给出老师发布的需求总数、小组的人数和小组中每名成员对团队的贡献值（如贡献值为 3，则该名成员有能力完成 3 个需求；贡献值可以为负，如某组员贡献值为 -2，则该组员不但完不成需求，还会导致其他组员少完成 2 个需求），请你计算出 Moca 小组最后能完成多少需求。

输入

第一行一个正整数 n ($4 \leq n \leq 6$) , 表示小组人数。

第二行 n 个整数 a_i ($-100 \leq a_i \leq 100$) , 表示第 i 个组员的贡献值。

第三行一个正整数 t ($1 \leq t \leq 500$) , 表示老师发布的需求总数。

输出

先输出一个**非负**整数, 表示 Moca 小组最后完成的需求数 (**依据实际, 这个需求数至少为 0, 至多为老师发布的需求数**) ; 如果 Moca 小组没有完成任何需求, 在下一行输出 `Moca finish 0 requirement!` ; 如果 Moca 小组完成了全部需求, 在下一行输出 `Moca finish all requirements!`。

如果你提交多次还不能对的话, 请至少看一下 `Hint`。

输入样例 1

```
4
40 8 -4 -4
100
```

输出样例 1

```
40
```

输入样例 2

```
5
10 10 10 10 -100
300
```

输出样例 2

```
0
Moca finish 0 requirement!
```

输入样例 3

```
6
10 10 10 10 10 -10
30
```

输出样例 3

30

Moca finish all requirements!

样例解释

样例 1 中，几人的贡献值和为 40，老师发布了 100 个需求，Moca 小组仅完成了其中 40 个。

样例 2 中，几人的贡献值和为 -60 ，然而在实际中需求完成的再少也不会少于 0 个，所以 Moca 小组完成了 0 个需求；即如果贡献值之和小于 0，则完成需求数为 0。

样例 3 中，几人的贡献值和为 40，然而老师仅发布了 30 个需求，所以 Moca 小组完成了全部的 30 个需求；即完成需求数大于老师发布的需求数，则完成需求数为**老师发布的需求数**。

Hint

程序主体框架：

```
if( sum <= 0 ){
    ...相应处理
}
else {
    if( sum >= need ){
        ...相应处理
    }
    else {
        ...相应处理
    }
}
```

其次，当你使用了 `sum` 保存贡献和时，如果 `sum` 定义在了 `main()` 内，请**初始化** `sum`，即 `sum = 0`。

最后，如果你真的非要写多个条件的话（虽然**这题用不到**），这么写是正确的语法：

```
if(( sum >= 0 ) && ( sum <= need )) {
    ...相应处理
}
```

std

```
#include <stdio.h>

int main()
{
    int sum = 0;
```

```

int n, a, need;
scanf("%d",&n);
for(int i = 1;i <= n;i++) {
    scanf("%d",&a);
    sum = sum + a;
}
scanf("%d",&need);

if( sum <= 0 ) {
    printf("%d\n",0);          // 根据题意，贡献值为负或者为 0 时是一个
需求也没完成的情况
    printf("Moca finish 0 requirement!");
}
else if( sum >= need ) {
    printf("%d\n",need);      // 根据题意，贡献值大于等于老师发布的需求
则完成了所有需求
    printf("Moca finish all requirements!");
}
else {
    printf("%d\n",sum);
}

return 0;
}

```

D 鸡兔同笼

题目描述

已知鸡和兔的总数量为 n , 总腿数为 m 。请你求出鸡的数目和兔的数目~

即：给定 n, m , 求以 x, y 为未知数的二元一次方程组 $\begin{cases} x + y = n \\ 2x + 4y = m \end{cases}$ 的**非负整数**解。

输入

不定组测试数据，最多不超过 100 组。（对于不定组输入，请注意看下面的提示。）

对于每组数据，一行两个整数，分别为 n 和 m ，且保证 $|n|, |m| \leq 10^8$ 。

输出

对于每组数据，输出一行，分别为鸡和兔的数量，中间用一个空格隔开。

若无非负整数解，则输出 `No answer`。

输入样例

```
14 32
10 16
```

输出样例

```
12 2
No answer
```

Hint

对于不定组输入，大家可以使用下面的模板。

```
while (scanf("%d%d", &n, &m) != EOF)
{
    //读取了两个int整数，分别存入变量n, m中
    //下面填写相应的处理代码
}
```

在电脑上运行时，每当你输入两个数，都会执行一次 `while` 内的代码块；当你不想再输入更多数据时，在 Windows 下可以按下 `Ctrl + Z`，在 Linux 或者 Mac 下可以按下 `Ctrl + D`，此时 `scanf` 函数将返回 `EOF`，停止继续将数据读入 `n` 和 `m`。

std

```
#include <stdio.h>

int main()
{
    int x, y, n, m;
    while (scanf("%d%d", &n, &m) != EOF)
    {
        x = (4 * n - m) / 2;
        y = n - x;
        if (m % 2)
            printf("No answer\n");
        else if (x < 0)
            printf("No answer\n");
        else if (y < 0)
            printf("No answer\n");
        else
            printf("%d %d\n", x, y);
    }
}
```

```
}  
}
```

E 分类三角形

题目描述

cyz 同学最近学习了三角形的相关概念，但他对定义把握的不是很清楚，希望你能帮助他。你只需要按如下规则来回答：

- 如果三条线段不能组成一个三角形，输出 `Not triangle`；
- 如果是直角三角形，输出 `Right triangle`；
- 如果是锐角三角形，输出 `Acute triangle`；
- 如果是钝角三角形，输出 `Obtuse triangle`；
- 如果是等腰三角形，输出 `Isosceles triangle`；
- 如果是等边三角形，输出 `Equilateral triangle`。

如果三角形同时满足以上多个条件，请依照上述顺序分别输出，用换行符分隔。

输入

输入一行三个正整数 a, b, c

输出

输出若干行字符串

输入样例 1

```
5 6 11
```

输出样例 1

```
Not triangle
```

输入样例 2

```
6 6 6
```


输出样例 2

```
Acute triangle
Isosceles triangle
Equilateral triangle
```

数据范围

保证 $1 \leq a \leq 1000, 1 \leq b \leq 1000, 1 \leq c \leq 1000$

std

```
#include<stdio.h>
int a,b,c;
int main()
{
    scanf("%d%d%d",&a,&b,&c);
    int t;

    if(a>b) t=a,a=b,b=t;
    if(a>c) t=a,a=c,c=t;
    if(b>c) t=b,b=c,c=t;

    if(a+b<=c) printf("Not triangle\n");
    else{
        if(a*a+b*b == c*c) printf("Right triangle\n");
        else if(a*a + b*b < c*c) printf("Obtuse triangle\n");
        else printf("Acute triangle\n");

        int tem = 0;
        if(a == b)
            tem = 1;
        if(b == c)
            tem = 1;
        if(tem == 1)
            printf("Isosceles triangle\n");
        if(a==b && b==c)
            printf("Equilateral triangle\n");
    }
}
```

题目描述

"Big Red Book" 发现了 xhesica 的爬虫,他们正在尝试找到 xhesica 然后 ban 掉他。

"Big Red Book" 给所有的请求返回了特别的数据包,里面有错误的图片 id, 如果 xhesica 试图用错误 id 拼接的 URL 爬去图片, "Big Red Book" 会直接封掉 xhesica 的 IP!

幸好 xhesica 发现了错误 id 的特点, 对于一个 id,如果其中某一位是数字 \$6\$, 那么它就是错误的。

你能帮 xhesica 判断一个 id 是否是正确的吗?

输入

一行, 一个非负整数 a , 代表 id,满足 $0 \leq id \leq 10^8$

输出

如果此 id 是错误的, 输出 Ban IP

如果此 id 是正确的, 输出 xhesica win

输入样例

132456

输出样例

Ban IP

样例解释

123456 中存在数字 6,所以输出 Ban IP

Hint

你在 C1 A题遇到过了什么奇怪的问题?

$\lfloor \frac{n}{1} \rfloor \bmod 10$ 可以获得个位上的数; $\lfloor \frac{n}{10} \rfloor \bmod 10$ 可以获得十位上的数, $\lfloor \frac{n}{100} \rfloor \bmod 10$ 可以获得百位上的数, 那么怎么获得千位上的数呢?

每次只需要取出某一位上的数字进行判断, 我们有没有什么办法通过循环来获取到每一位上的数字呢?

std

```
#include<stdio.h>
int x;
int main(){
    scanf("%d",&x);
    while(x){
        if(x%10==6){
            puts("Ban 1P");
            return 0;
        }
        x/=10;
    }
    puts("Xhesica win");
    return 0;
}
```

G yet another a+b problem

题目描述

输入三个整数 a, b, p , 希望你找出 c 满足 $a + b \equiv c \pmod{p}$, $0 \leq c < p$ 。

即找到一个位于 0 和 $p - 1$ 之间的数 c (包括 0 和 $p - 1$), 使得 $a + b - c$ 是 p 的倍数。

输入

一行, 三个整数 a, b, p , $-10^9 \leq a, b \leq 10^9, 1 \leq p \leq 10^9$ 。

输出

一行, 一个整数 c 。

输入样例 (1)

5 8 7

输出样例 (1)

6

输入样例（2）

```
3 -5 6
```

输出样例（2）

```
4
```

std

```
#include<stdio.h>
int main(){
    int a,b,p;
    scanf("%d%d%d",&a,&b,&p);
    int c = a+b;
    printf("%d",(c%p+p)%p);
}
```

H shtog 挑武器

题目描述

shtog来到了冒险者大陆，在出发冒险前他要去商店买一个尽可能厉害的武器来防身，你能帮他找出武力值最高的武器吗？

输入

第一行一个整数 n ($1 \leq n \leq 10^5$)，表示商店的武器数量。

接下来一行 n 个正整数（均在 `int` 范围之内），整数之间用空格隔开（末尾无多余空格），第 i 个整数代表编号为 i 的武器的武力值

输出

输出一个整数，表示武力值最高的武器的编号（如果有武力值一样高的武器则优先输出编号大的）。

输入样例

```
3
1 22 4
```

输出样例

```
2
```

std

```
#include <stdio.h>

int main() {
    int n, i;
    int m, mi = -1;
    scanf("%d", &n);
    for (i = 1; i <= n; ++i) {
        int t;
        scanf("%d", &t);
        if (i == 1) {
            m = t;
            mi = 1;
        } else {
            if (t > m) {
                m = t;
                mi = i;
            } else if (t == m) mi = i;
        }
    }
    printf("%d\n", mi);
    return 0;
}
```

I 小 P 的乌龟爬水井 2

题目描述

小 P 在遇到了这样一道题：水井高 500 厘米，井底有一只乌龟，白天其可以向上爬 50 厘米，但晚上会回落 30 厘米，问其到第几天才能爬到井口。

小 P 觉得这题过于简单，于是将其中的数字全换成英文字母，即题目为：井高 h 厘米，井底有一只乌龟，白天其可以向上爬 a 厘米，但晚上会回落 b 厘米，问其到第几天才能爬到井口。

小 P 觉得这题还是很简单，于是想把 a, b 的范围扩展到 0 及 0 以下。对应的，回落负数厘米代表上升相应厘米，上升负数厘米代表下降相应厘米。

注意：题目中认为一天为一个白天加一个夜晚，即使乌龟在当天夜晚的最后时刻到达井口，也认为是在这天到达；乌龟最低位于井底，不会位于更低的地方。

现在，小 P 将这道题交给了你，请你编写一段程序，从而完成此题。

输入

第一行给出整数 T ，代表数据组数。

接下来每组数据中，将给出三个整数，分别为 h, a, b 。

具体含义见题目描述。

输出

一行整数，代表乌龟第几天才能爬到井口。

如果其永远无法爬到井口，请输出 `Impossible`。

每组输出之间用换行间隔。

输入样例 1

```
5
500 50 30
500 50 50
500 50 -30
500 50 -50
500 600 700
```

输出样例

```
24
Impossible
7
5
1
```

数据范围

对于 30% 的测试样例， $0 < T \leq 100$ ，同时 $a \geq 0$ 且 $b \geq 0$

对于 100% 的测试样例, $0 < T \leq 2 \times 10^5$, $0 < h \leq 10^9$, $-10^9 \leq a, b \leq 10^9$ 。

Hint

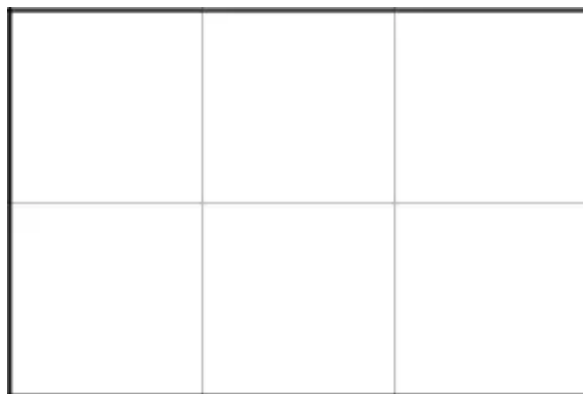
可以参考一下 C1 的 H

std

```
#include<stdio.h>
#include<string.h>
#include<math.h>
#include<stdlib.h>
#include<time.h>
signed main()
{
    int a,b,h,ans,a1,b1;//此题中int就足够了
    scanf("%d%d%d",&h,&a1,&b1);
    a = a1, b = b1;
    if(a1 < 0 && b1 < 0)
        a = -b1, b = -a1;
    if(a1 < 0)
        a = 0, b = b1 - a1;
    if(b1 < 1)
        b = 0, a = a1 + b1;
    if(a>=h)//a>=h时, 第一天白天就可到顶, 与b无关
    {
        printf("1\n");
    }
    else if(a<=b)//a<h且a<=b时, 总无法到顶
    {
        printf("Impossible\n");
    }
    else//b<a<h时, 可通过表达式计算出天数
    {
        ans=((h-a)+(a-b-1))/(a-b)+1;//a/b向上取整等价于(a+b-1)/b的向下取
        printf("%d\n",ans);
    }
    return 0;
}
```

题目描述

在一次探险之旅中，探险家水獭发现了一片巨大的网格。这个网格由许多大小为 1×1 的正方形组成，左下角为 $(0, 0)$ ，而右上角为 (m, n) 。下图展示了 $m = 3$ ， $n = 2$ 的情况：



探险家水獭每天都会从营地 $(0, 0)$ 出发，选择一个新的网格点进行探索（不包括 $(0, 0)$ ），然后再返回营地。水獭发现，每次沿着一条直线前往某个网格点时，路途中可能会经过若干个横纵坐标都为整数的点。

水獭的目标是探索整个 $m \times n$ 的网格区域，即 $(m + 1) \times (n + 1) - 1$ 个不同的网格点。水獭非常好奇，当他完成所有的探险任务后，累积前往这些点的途中总共会经过多少个整数点（**注意是途中，不包括两个端点**）。

水獭知道这个问题不简单，于是他找到了会编程的 Moca 来帮忙编程解决这个谜题。

输入

两个正整数，分别代表 m 和 n ，含义如题目所示，保证 $1 \leq m \leq 1000$ ， $1 \leq n \leq 1000$

输出

一个正整数，表示探险家水獭完成所有的探险任务后，累积前往这些点的途中总共会经过多少个整数点

输入样例

3 2

输出样例

5

样例解释

当探索 $(2, 0)$ 时，会经过 $(1, 0)$ 一个整数点。

当探索 (3,0) 时, 会经过 (1,0) 和 (2,0) 两个整数点。

当探索 (0,2) 时, 会经过 (0,1) 一个整数点。

当探索 (2,2) 时, 会经过 (1,1) 一个整数点。

其它情况不会经过整数点。

综上, 探索其余十一个点的过程中, 一共会经过 5 个整数点。

std

```
#include<stdio.h>
int main()
{
    int m,n,gcd,ans,temp,a,b;
    scanf("%d%d",&m,&n);
    for(int i=1;i<=m;i++) {
        gcd = 0,temp = 0;
        for(int j = 1;j <= n;j++) {
            a = i,b = j;
            while(b != 0) {
                temp = a % b;
                a = b;
                b = temp;
            }
            ans += a-1;
        }
    }
    ans += (m*(m-1)+n*(n-1))/2;
    printf("%d",ans);
    return 0;
}
```