

C1 - Statement

A 教师节快乐

题目描述

今天是我们程设的第一节课，也是我国的第 40 个教师节。

18 年一路走来，我们的成长过程中，遇到许多良师，此时此刻，让我们大声表白一下吧！

输入

无

输出

一行，如下所示的字符串

```
Dear teacher, I love yOu.
```

输入样例

无

输出样例

```
Dear teacher, I love yOu.
```

Hint

老师跟你开了一个玩笑（如果你还这么不仔细，难怪受到老师批评了），请仔细查看输出内容！多次输入不对，为什么不试试直接拷贝字符串呢？

可以参考课件 P1 例 1-1

Author : Prof. Song

std

```
#include <stdio.h>

int main() {
    printf("Dear teacher, I love you.");
    return 0;
}
```

B 献给老师的小红花

题目描述

刚喊完老师辛苦了，现在是时候送出你悄悄准备的小红花啦！

为了表达对老师们辛勤付出的感谢，同学们精心准备了 m 朵小红花，想要送给到场的每一位老师。同学们希望每位老师都能分到同样多的鲜花，所以可能会有剩下的鲜花。

此刻，已经有 n 位老师到场。聪明的你能帮同学们算一下，如果这些小红花只分给目前在场的老师，每位老师最多能分到多少朵？剩下的又有多少呢？当然，如果还没有老师到场（即 $n = 0$ ），同学们也会耐心等待，期待老师们的到来。

输入

输入包含两个正整数 n 和 m ，分别表示到场的老师人数和准备的小红花数。保证 $0 \leq n \leq 10^8$ ， $0 \leq m \leq 10^8$ 。

输出

如果还没有老师到场（即 $n = 0$ ），输出 `wait a minute`。

否则，输出两行。第一行是每名老师最多能拿到的小红花数，第二行是剩下的小红花数。

输入样例 1

```
2 5
```

输出样例 1

```
2
1
```

输入样例 2

输出样例 2

```
wait a minute
```

Hint

可以参考课件 P1 例 1-1-a、例 1-3 和 例 1-6

std

```
#include <stdio.h>

int main() {
    int n, k;
    scanf("%d%d", &n, &k);
    if(n == 0) {
        printf("wait a minute\n");
    }
    else {
        printf("%d\n", k / n);
        printf("%d\n", k % n);
    }
    return 0;
}
```

C 拼接 URL

题目描述

Xhesica 正在学习如何写爬虫爬取 "Big Red Book "上的图片, 他得到了目标图片的id , 一个正整数 a , 希望把它加到 `https://sns-img-bd.brdcdn.com/` 这个 URL 的后面来得到目标 URL , 你能够帮助他吗?

输入

一行, 一个正整数 a , 满足 $1 \leq a \leq 10^6$

输出

一行，代表你拼接的 URL。

输入样例

```
100
```

输出样例

```
https://sns-img-bd.brdcdn.com/100
```

样例解释

拼接 `https://sns-img-bd.brdcdn.com/` 和 `100` 得到 `https://sns-img-bd.brdcdn.com/100`

Hint

可以参考课件 P1 例 1-2

```
#include<stdio.h>

int main(){
    int a;
    scanf("%d",&a);
    printf("https://sns-img-bd.brdcdn.com/%d",a);
    return 0;
}
```

D 摩卡与分数统计

题目描述

某次上机中，Moca 需要对同学们的上机分数进行统计。

在这次上机中，总共有三道编程题，每道题的满分是 40 分，三道题总分为 120 分。然而，由于 100 分即可获得满分，学校规定任何超过 100 分的总成绩将被记作 100 分。

现在，Moca 需要根据某名学生在三道题上的得分来计算他的最终总成绩。Moca 希望能够通过程序自动计算学生的总成绩，你能写一个程序帮助她吗？

输入

一行，三个正整数，表示某同学三道题目的得分；每道题得分 s ，有 $0 \leq s \leq 40$

输出

一个正整数，表示该同学的总成绩。

输入样例 1

```
10 20 30
```

输出样例 1

```
60
```

输入样例 2

```
40 35 35
```

输出样例 2

```
100
```

样例解释

在第一组样例中，学生的总得分为 60 分，未超过 100 分，因此最终成绩为 60 分。

在第二组样例中，学生的总得分为 110 分，超过 100 分，因此最终成绩记为 100 分。

Hint

可以参考课件 P1 例 1-2 和 例 1-5

std

```
#include <stdio.h>

int main() {
    int sum = 0;
    int a, b, c;
    scanf("%d%d%d", &a, &b, &c);

    sum = a + b + c;

    if(sum >= 100) {
        printf("100");
    }
}
```

```
    }  
    else {  
        printf("%d", sum);  
    }  
  
    return 0;  
}
```

E 小懒獭与叉积

题目描述

小懒獭上大学了，正在学习高等代数，很快会学习到一种重要的向量运算，叫做叉积。你能帮助她写一个计算器，用来计算两个三维向量的叉积吗？

两个三维向量 $\vec{a} = (a_1, a_2, a_3)$, $\vec{b} = (b_1, b_2, b_3)$ 的叉积结果为：

$$\vec{a} \times \vec{b} = (a_2b_3 - a_3b_2, a_3b_1 - a_1b_3, a_1b_2 - a_2b_1)$$

输入

两行，第一行三个正整数表示 a_1 、 a_2 和 a_3 ；第二行三个正整数表示 b_1 、 b_2 和 b_3 ，保证所有输入数的绝对值不超过 1000。

输出

一行三个整数，表示 \vec{a} 与 \vec{b} 叉积的三个分量

输入样例

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

输出样例

```
-4 8 -4
```

std

```
int main() {
    int sum = 0;
    int a1, a2, a3;
    int b1, b2, b3;
    scanf("%d%d%d", &a1, &a2, &a3);
    scanf("%d%d%d", &b1, &b2, &b3);

    printf("%d ", a2 * b3 - a3 * b2);
    printf("%d ", a3 * b1 - a1 * b3);
    printf("%d", a1 * b2 - a2 * b1);

    return 0;
}
```

F 小水獭与月饼

题目描述

中秋节快到了，小水獭们却为手中的月饼分配感到苦恼。

每只小水獭手上都有不同数量的月饼，有的水獭手上有很多月饼，而有的却只有寥寥几块，甚至还有一些小水獭手上没有月饼。小水獭们知道，好东西要大家一起分享，于是小水獭们想重新分配手上的月饼，让所有水獭拿到的月饼尽可能平均；即重新分配后，任意两只水獭手中的月饼数差距不能超过一块。

小水獭们为这个难题而感到困扰，于是向你求助，你能帮帮水獭们吗？

输入

输入包含若干行，每行一个整数 k_i ，如果 $k_i \neq -1$ ，表示第 i 只水獭手中的月饼数量，保证此时 $0 \leq k_i \leq 1000$ ；如果 $k_i = -1$ ，则表示输入结束。

保证输入不超过 1000 行，至少有一只水獭。

输出

输出一个正整数，表示重新分配后，拥有月饼数最少的水獭手中的月饼数。

输入样例 1

```
1
2
3
-1
```

输出样例 1

```
2
```

输入样例 2

```
3
3
4
-1
```

输出样例 2

```
3
```

样例解释

在第一组样例中，三只小水獭一共有六块月饼，重新分配后，三只水獭每只两块月饼，拥有月饼数最多的水獭手中有两块月饼。

在第二组样例中，三只小水獭一共有十块月饼，重新分配后，两只水獭手上三块月饼，一只水獭手上四块月饼，拥有月饼数最少的水獭手中有三块月饼。

Hint

可以参考课件 P1 例 1-8

std

```
#include <stdio.h>

int main() {
    int sum = 0, n = 0; // 表示月饼总数和水獭的数目
    int num;           // 表示读入的当前水獭的月饼数

    scanf("%d", &num); // 输入第一个月饼数

    while(num != -1) {
        sum = sum + num; // 月饼累加
    }
}
```



```

        n = n + 1;           // 水獭数累加
        scanf("%d", &num);   // 输入下一个月饼数
    }

    printf("%d", sum / n);    // 两个正整数之间除法向下取整

    return 0;
}

```

G 摩卡与水獭餐厅

题目描述

Moca 开了一个水獭餐厅。

餐厅里有 k 个座位排成一行，每当一只新的小水獭来用餐，Moca 总是从左到右、按顺序将他们安排到空座位上。

今天有 n 只小水獭来用餐。Moca 希望你能帮忙安排这些水獭的座位。当然，由于餐厅的生意太过火爆，来的水獭数可能多于座位数，无法使全部水獭入座，Moca 就只能遗憾地告诉水獭们 `Too many little otters!`。

输入

两个正整数 $n\ k$ ，分别代表表示水獭的个数和座位数。保证 $1 \leq n, k \leq 100$

输出

如果小水獭们无法全部入座，输出 `Too many little otters!`。

否则，用 `*` 表示有水獭入座的座位，用 `_` 表示空座位，输出餐厅的入座情况；两个座位之间用一个空格隔开

输入样例 1

```
7 10
```

输出样例 1

```
* * * * * * * _ _ _
```

输入样例 2

输出样例 2

```
Too many little otters!
```

Hint

可以参考课件 P1 例 1-7

std

```
#include <stdio.h>

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

    if(n > k)
        printf("Too many little otters!");
    else {
        for(int i = 1; i <= n ;i++) {
            printf("* ");
        }
        for(int i = n + 1; i <= k ;i++) {
            printf("_ ");
        }
    }

    return 0;
}
```

H 小 P 的乌龟爬水井

题目描述

小P在网上看到了一道题：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，问其最早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。因此，最终的题目为：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，问其最早第几天才能爬到井口。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。

注意：题目中的天数从第 1 天开始计算；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。

输入

一行，三个整数，分别为 h, a, b 。其中， $0 \leq h, a, b \leq 10^9$ 。

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`。

输入样例 1

```
500 50 30
```

输出样例 1

```
24
```

输入样例 2

```
500 50 50
```

输出样例 2

```
Impossible
```

输入样例 3

```
500 600 700
```

输出样例 3

```
1
```

样例解释

对于第一组数据，前 23 天乌龟走到了 460 cm处，在第 24 天白天走 50 cm后走到井口。

对于第二组数据，乌龟每天上升 50 cm，下降 50 cm，故乌龟在每天结束时始终在井底，不可能到井口。

对于第三组数据，虽然乌龟每天上升距离小于下降距离，但乌龟可以在第一天白天直接到达井口。

std

```
#include<stdio.h>
#include<string.h>
#include<math.h>
#include<stdlib.h>
#include<time.h>
signed main()
{
    int a,b,h,ans;//此题中int就足够了
    scanf("%d%d%d",&h,&a,&b);
    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
整
        printf("%d\n",ans);
    }
    return 0;
}
```

I 军乐团破冰

题目描述

BUAAers 最喜欢军训啦！

军乐团总指挥「Ormandy」先生正在为破冰聚餐购买奶茶。信息大类中一共有 8 个连队有人参加军乐团，但这 8 个连队各自的参加人数不尽相同，「Ormandy」先生希望买来的奶茶能完美均分。

完美均分需同时满足以下 3 个条件：

- 所有奶茶都分配到同学手中，无剩余。
- 每个连队分配到的奶茶总数相同。
- 同一连队的每个同学分配到的奶茶杯数相同且不为 0。

你能帮「Ormandy」先生求出完美均分至少要买几杯奶茶嘛~（总指挥本人是不喝奶茶的）

输入

一行 8 个整数，分别为 a_1, a_2, \dots, a_8 ，其中 a_i 表示第 i 个连队的参加人数，保证 $1 \leq a_i \leq 200$ 。

输出

一个整数，为达成完美均分至少需要购买的奶茶杯数，题目保证结果在 `int` 范围内。

输入样例

```
10 20 15 25 5 30 20 15
```

输出样例

```
2400
```

样例解释

每个连队 300 杯奶茶，一至八连每个同学的奶茶杯数分别为 30, 15, 20, 12, 60, 10, 15, 20，此时刚好达成完美均分。

HINT

a 和 b 的最小公倍数 $\text{lcm}(a, b) = \frac{a \times b}{\text{gcd}(a, b)}$

a 、 b 和 c 的最小公倍数 $\text{lcm}(a, b, c) = \text{lcm}(\text{lcm}(a, b), c)$

??? 你问什么是 gcd ？，去 PPT 例 9 找找答案叭~

本题虽然保证结果一定在 `int` 范围内，但也要小心在运算过程中不要超出 `int` 范围哦~

拓展知识

`int` 型变量能存储的整数范围是 $[-2147483648, 2147483647]$ ，存储超过这个范围的整数就会溢出，变为错误的数字。

例如运行一下如下代码，看看会输出什么结果呢？

```
#include<stdio.h>
int main()
{
    int a=100000,b=100000,c=100;
    int d=a*b/c;
    int e=a/c*b;
    printf("a*b: %d\n",a*b);
    printf("d:    %d\n",d);
    printf("e:    %d\n",e);
    return 0;
}
```

std

```
#include<stdio.h>
int main()
{
    int lcm,a;
    scanf("%d",&lcm); //第1个数
    for(int i=1;i<=7;i++)
    {
        scanf("%d",&a); //后7个数
        for(int gcd=200;gcd>=1;gcd--) //从高向低枚举gcd
        {
            if(lcm%gcd==0&&a%gcd==0) //找到gcd了
            {
                lcm/=gcd; //更新lcm,注意此处写法!!!
                lcm*=a;
                break; //循环提前结束
            }
        }
    }
    printf("%d",lcm*8); //别忘了最后*8
    return 0;
}
```

题目描述

探险家水獭在探索远古遗迹时发现了一组古老的神秘数列。传说破解这个数列能够揭示世界的终极秘密。探险家水獭急切地想解开这些谜团，但每次解开一个数字，都需要动用大量的智慧和运算技巧。于是，他决定向会编程的你求助。

根据古老的记载，数列的第一个数字已经被解开—— $a_1 = 1$ ，但接下来的数字却隐藏在一串复杂的递推公式中：

$$a_n = \max\{a_k + a_{n-k} + k(n-k) \mid 1 \leq k < n\}$$

探险家水獭想知道数列中第 m 项的值，你能帮他解答吗？

输入

一个正整数 m ，表示要求出数列的第 m 项，保证 $1 \leq m \leq 40000$ 。

输出

一个正整数，表示 a_m 的值。

输入样例

2

输出样例

3

std

```
#include <stdio.h>

int main() {
    int n;
    scanf("%d", &n);
    printf("%d", n * (n + 1) / 2);

    return 0;
}
```