A 等式填空

时间限制：1000ms　内存限制：65536kb

通过率：1477/1599 (92.37%)　正确率：1477/4459 (33.12%)

题目描述

对于给定的一个某些数位被遮挡住的两位数乘法等式，请问有多少种填空方案使得这个等式成立？

等式的格式如下：

$$a\_*\_b=\_cd\_$$

其中 a、b、c、d 为给定的数字，a 代表第一个运算数的十位，b 代表第二个运算数的个位，c,d 分别代表结果的百位和十位。下划线的位置是需要填空的数位，可以分别填充一个 0~9 的数码。
对于一个填空方案 e,f,g,h，若满足 $ae*fb=gcdh$，则称为一个可行的填空方案。

输入

多组数据。

第一行为一个正整数 $T(T<10)$，为数据的组数。

对于每组数据，输入一行四个整数 $a,b,c,d(0\le a,c,b,d\le 9)$，含义见题目描述。

输出

对于每组数据，若没有可行的填空方案，则输出一行 IMPOSSIBLE!（注意感叹号）。
否则输出若干行，每行按 e,f,g,h 的字典序来输出一个可行的填空方案，输出格式见样例，注意**冒号后有一个空格**，且等式直接按 $ae*fb=gcdh$ 的格式输出即可，不需要忽略前导 0。

输入样例

```
3
1 1 1 1
9 8 4 3
6 9 5 3
```

输出样例

```
case1: 10*11=0110
IMPOSSIBLE!
case1: 65*39=2535
case2: 66*99=6534
```

样例解释

对于第一组数据，仅有一个等式 $10*11=0110$ 满足，输出一行；
对于第二组数据，无法满足等式，输出 IMPOSSIBLE!；
对于第三组数据，有两个满足条件的等式，按照格式输出两行。

考察知识点

for 循环

难度系数 3

解题思路

这道题是一道十分基础的枚举题。为了满足字典序的要求，e,f,g,h 都应从 0 开始枚举到 9，且枚举顺序（循环顺序）应该是 e,f,g,h，每枚举一个方案判断是否满足等式，是就按格式输出，并将记录填空方案数量的变量加 1。枚举结束后若记录到的填空方案数为 0，则输出 IMPOSSIBLE!

参考代码

```
#include <stdio.h>

int main()
{
    int T;
    scanf("%d", &T);
    while (T --) {
        int a, b, c, d, e, f, g, h;
        scanf("%d%d%d%d", &a, &b, &c, &d);
        int cnt = 0;
        for (e = 0; e < 10; ++ e) {
            for (f = 0; f < 10; ++ f) {
                for (g = 0; g < 10; ++ g) {
                    for (h = 0; h < 10; ++ h) {
                        if ((a * 10 + e) * (f * 10 + b) == (1000 * g + 100 * c +
10 * d + h)) {
                            printf("case%d:  %d%d*%d%d=%d%d%d%d\n",  +
+ cnt, a, e, f, b, g, c, d, h);
                        }
                    }
                }
            }
        }
        if (!cnt) {
```

```
                printf("IMPOSSIBLE!\n");
            }
        }
        return 0;
}
```

B 质因数分解

时间限制：1000ms　内存限制：65536kb

通过率：1449/1560 (92.88%)　正确率：1449/4488 (32.29%)

题目描述

给定一个由**两个质数**相乘得到的正整数 n，请求出较大的那个质数（如果二者相等，也只输出一次）。

输入

多组数据。

第一行为一个正整数 T(T<10)，为数据的组数。

对于每组数据，为一行一个正整数 n(0<n≤2·10e9)，为待分解的整数。

数据保证 n 为两个质数相乘得到。

输出

对于每组数据，输出一行一个整数 p，为较大的那个质数。

输入样例

```
2
25
91
```

输出样例

```
5
13
```

考察知识点

枚举

难度系数 2

解题思路

本题其实就是要找出 n 的因数，第一想法可能是从 1 到 n 遍历一遍，判断是否能整除 n，如果能就是 n 的因数，之后取大的输出即可。
但是 n 最大是 2*10^9，而计算机一秒运行的次数大概是 10^8，所以遍历一遍是会超过时间限制的。
考虑 n 的一个因数分解 n=a*b，假设 a<=b，那么显然有 a 小于等于根号 n，于是我们可以把遍历的范围缩减到 1 到根号 n，找到一个可整除 n 的数 a 就找到了那个较小的因数，那只要输出 n/a 即可。
另外，除了 2 以为的质数都是奇数，可以进一步缩短枚举次数。

参考代码

```
#include <stdio.h>

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

    while (T --) {
        int n, i;
        scanf("%d", &n);

        if (n % 2 == 0) {
            printf("%d\n", n / 2);
            continue;
        }
        for (i = 3; i * i <= n; i += 2) {
            if (n % i == 0) {
                printf("%d\n", n / i);
                break;
            }
        }
    }
    return 0;
}
```

C 求数列的一项

时间限制：1000ms　内存限制：65536kb

通过率：1506/1597 (94.30%)　正确率：1506/4890 (30.80%)

题目描述

数列 $A=1,1,3,5,11,21\ldots$ ，其中 $a_i=2\cdot a_{i-2}+a_{i-1}(3\le i)$。数列下标从 1 开始编号。

请你求出这个数列的第 n 项。

输入

输入一行，一个整数 $n(n\in[1,50])$。

输出

一个整数,代表 a_n（数列的第 n 项）。答案的数值可能较大，需要用到 long long（输出请用 %lld ）。

输入样例

```
4
```

输出样例

```
5
```

考察知识点

for 循环

难度系数 3

解题思路

每一项都只跟前两项有关,可以通过保存前两项的值,求出第三项,再舍弃掉前一项的值,通过第二项和目前求出的第三项,求出第四项的值,如此循环操作得到第 n 项.

(由于答案数据较大,要使用 long long 存储数列的值)

代码

```
#include <stdio.h>

int main()
{
    int n, i;
    scanf("%d", &n);
    long long ai, ai_2, ai_1;
    if (n < 3) {
        ai = 1;
    } else {
        ai_2 = 1;
        ai_1 = 1;
        for (i = 3; i <= n; ++ i) {
            ai = 2 * ai_2 + ai_1;
            ai_2 = ai_1;
            ai_1 = ai;
        }
    }
    printf("%lld\n", ai);
    return 0;
}
```

D 破译密码

时间限制：1000ms　内存限制：65536kb

通过率：1328/1424 (93.26%)　正确率：1328/2836 (46.83%)

题目描述

通信员小 A 截获了帝国通信的密码字符串。已知原密码是字符串中每个字符 ASCII 码加上 4 的字符表示，现在请你帮他还原原本的密码。

输入

输入一行，包括一个字符串（可以用 %c 读入单个字符，或者用 %s 读入整个字符串，读入的字符串将以 '\0' 结束），保证输入串长度不超过 100。

注意输入样例的末尾有一个特殊字符（ASCII 码为 96）。

输出

输出一行，还原之后的字符串。

输入样例

```
Dahhksknh`
```

输出样例

```
Helloworld
```

考察知识点

for 循环

ASCII 码

难度系数 3

解题思路

要把字符串中每一个字符都都通过 ASCII 码进行+4 来得到最终输出的结果,使用 for 循环每一位+4 然后输出每一位字符.

代码:

```
#include <stdio.h>

int main()
{
    char c;
    while (scanf("%c", &c) != EOF) {
        printf("%c", c + 4);
    }
    return 0;
}
```

E 阿狄的冒险

时间限制：1000ms　内存限制：65536kb

通过率：673/830 (81.08%)　正确率：673/2416 (27.86%)

题目描述

阿狄最近沉迷探索神庙，通关神庙需要特定的素材帮助。每次需要某种素材时，阿狄会首先在自己的背包中寻找；若不存在，则需要返回仓库取出该种素材使用，并放入背包以备后面的冒险。

背包有 M 个空格用以存放素材，当背包中有多余空格（素材种类数小于 M）时会直接放入背包；若背包已满，则会将最早放入背包的素材放回仓库以腾出空间，存放所需新素材。这次阿狄有 N 个神庙需要探索，出发时背包是空的，那么他需要返回多少次仓库？

输入

输入共两行。

第一行为两个整数 M 和 N，代表背包上限和神庙总数。
第二行为空格分隔的 N 个整数，依次代表神庙所需的素材种类标号。保证标号在 [0,1000] 之间。
$0 \le M,N , M \le 100 , N \le 1000$。

输出

输出一个整数，表示需要返回仓库的次数。

输入样例

```
2 5
1 2 1 3 1
```

输出样例

```
4
```

HINT

第 1 个神庙，素材 1 不在背包，返回仓库并放入背包。
第 2 个神庙，素材 2 不在背包，返回仓库并放入背包。
第 3 个神庙，素材 1 在背包中，继续冒险。
第 4 个神庙，素材 3 不在背包，背包已满，将最早放入的素材 1 清除，返回仓库并把素材 3 放入背包。
第 5 个神庙，素材 已不在背包，背包已满，将最早放入的素材 2 清除，返回仓库并把素材 1 放入背包。

这次的题意有一些纰漏，没有解释清楚给大家带来困扰很抱歉。
题意是应该是模拟取拿的过程中当背包中没有该素材的情况，首先使用该素材再将素材放入背包，所以当背包为 0 的时候会每一次寻找素材都需要返回仓库。题中另一个坑点在于

素材的编号可能为 0，如果直接初始化背包为 0 来表示空背包状态也会 WA。
解题可以使用两个循环，每次寻找素材时遍历当前背包，设立一个标志位表示背包的尾部通过取模循环背包的取放。因为素材种类数量较少，也可以更简单的单独设置标志表示每个素材是否在背包中，两个头尾标志实现取放过程，下面是实现代码

```
#include <stdio.h>

#define MAXN (1010)

int head, tail, bag[MAXN];
int has[MAXN];

int main()
{
    int m, n, i, ans = 0;
    scanf("%d%d", &m, &n);

    head = tail = 0;
    for (i = 0; i < n; ++ i) {
        int x;
        scanf("%d", &x);

        if (!has[x]) {
            ++ ans;
            has[x] = 1;
            bag[tail ++] = x;
            if (tail - head > m) {
                has[bag[head ++]] = 0;
            }
        }
    }
    printf("%d\n", ans);
    return 0;
}
```

F Terry 的理财计划

时间限制：1000ms　内存限制：65536kb

通过率：717/895 (80.11%)　正确率：717/3854 (18.60%)

题目描述

Terry 的零花钱一直都是自己管理。每个月的月初，妈妈给 Terry 300 元钱，Terry 会预算这个月的花销，并且总能做到实际花销和预算的相同。
Terry 可以随时把**整百**的钱存在妈妈那里，到了年末她会加上 20% 还给 Terry 。因此 Terry 制定了一个计划：每个月的月初，在得到零花钱后，如果他预计到这个月的月末手中还会有多于 100 元或恰好 100 元，他就会把手中扣除掉预算之后的整百的钱存在妈妈那里，剩余的钱留在自己手中。

Terry 发现这个计划的主要风险是，存在妈妈那里的钱在年末之前不能取出。而有可能在某个月的月初，Terry 手中的钱加上这个月的零花钱，不够这个月的原定预算。如果出现这种情况，Terry 将不得不在这个月省吃俭用，压缩预算。

现在请你根据某年 1 月到 12 月每个月 Terry 的预算，判断会不会出现这种情况。如果不会，计算到这年年末，妈妈将 Terry 平常存的钱加上 20% 还给 Terry 之后， Terry 手中会有多少钱。

输入

12 行，每行包含一个小于 350 的非负整数，分别表示 1 月到 12 月 Terry 的预算。

输出

一行，这一行只包含一个整数。

如果计划实施过程中出现某个月钱不够用的情况，输出 -X，X 表示出现这种情况的第一个月。
否则输出到年末 Terry 手中会有多少钱。

输入样例

```
300
300
300
300
300
300
300
300
300
300
300
100
```

输出样例

```
240
```

考察知识点

for 循环

难度系数 3

解题思路：
每个月开始的时候，先检查是否出现预算不够的情况，如果出现了就记录第一次出现的月。
本题的所谓存钱，就是将手中的整百的钱上交。
十二个月过后，看是否出现预算不够的情况，如果出现就将月份取负输出。
否则，将存款*1.2 再加上手中的零钱 然后输出。
本题只要注意审题，模拟整个过程并不是很困难。

标程：
```
#include<stdio.h>
int main() {
    int salary,b,pin,d,handing,banking;
   handing=0;
   banking=0;
    pin=0;
    d=13;
    for(b=1;b<=12;b++) {
        scanf("%d",&salary);
        if(handing+300<salary) {
            pin=1;
         d=(d<b)?d:b;
        }
        else {
            banking=banking+(300+handing-salary)/100*100;
           handing=(handing+300-salary)%100;
        }

    }
    if(pin==1) {
        printf("-%d\n",d);
    }
    else {
        banking=banking*12/10;
        printf("%d",banking+handing);
    }
    return 0;
}
```

G 傻傻 Aqi 的猜素数程序

时间限制：1000ms　内存限制：65536kb

通过率：1281/1447 (88.53%)　正确率：1281/5721 (22.39%)

题目描述

Alice 和 Aqi 在玩猜哪个数是素数的游戏，猜的正确次数多的就可以指使对方干一周活，懒懒 Aqi 才不想输呢。

他灵机一动想到可以编一个程序，但他编程实在太不好了，你们能帮他吗？

（如果你们不快点编的话，聪明的 Alice 就要编好了哟~

给定一个整数 n，要求判断其是否为素数（0,1 以及负数均不属于质数）。

输入

多组数据。

每组数据，输入一行，一个整数 n（保证在 int 范围内）。

输出

对于每组数据，若是素数输出 yes，否则输出 no（注意大小写）。

输入样例

```
23
9
1
```

输出样例

```
yes
no
no
```

HINT

如果你用枚举每个数是否能整除 n 的方法判断的话，我们的枚举上界其实不需要到 n−1 的。（至于枚举上界设到哪里枚举得即全面又最少，就交给聪明的大家自己去想啦~

考察知识点

素数、循环

难度系数 3

解题思路

素数定义为在大于 1 的自然数中，除了 1 和它本身以外不再有其他因数的数字。所以我们可以用枚举法，依次枚举每个数是否能整除 n。但是我们的枚举上界只需要设置到 $\sqrt{n}+1$ 就可以了，因为如果到此为止都没有可以整除的数的话，此后也都不会有了。

代码

```
#include <stdio.h>

int judgePrime(int n)
{
    if (n == 2) {
        return 1;
    }
    if (n < 2 || n % 2 == 0) {
        return 0;
    }
    int i;
    for (i = 3; i * i <= n; ++ i) {
        if (n % i == 0) {
            return 0;
        }
    }
    return 1;
}

int main()
{
    int n;
    while (scanf("%d", &n) != EOF) {
        printf(judgePrime(n) ? "yes\n" : "no\n");
    }
    return 0;
}
```

H 傻傻 Aqi 与欧几里得算法求 GCD

时间限制：1000ms　内存限制：65536kb

通过率：1276/1426 (89.48%)　正确率：1276/4776 (26.72%)

题目描述

傻傻 Aqi 想编程求整数 a,b 的最大公约数（GCD，greatest common divisor）。

他苦思冥想后决定用遍历的方法，这个朴素的思路是正确的，但是当 a,b 数值较大时（如 10^7 ），该算法的时间复杂度较高。

这时聪明的 Alice 告诉他，有一种算法叫欧几里得算法，首先可证明 (a,b) 的最大公约数同时也是 (b,a mod b) 的最大公约数，那么我们得到求解 (a,b) 的 GCD 的算法如下：

- 若 (a,b) 其中之一为 0，则最大公约数为非 0 的那个。

- 若 (a,b) 均不为 0 ，则继续用此算法求解 (b,a mod b) 的 GCD。

Alice 都说的这么明白了，可是傻傻 Aqi 还是不会编，你们能帮他吗？

输入

多组数据,。

每组输入两个非负整数 a,b（输入保证 a,b 不同时为 0，且 a,b 在 int 范围内）。

输出

对于每组输入，输出其最大公约数（GCD），保证在 int 范围内。

输入样例

```
54 36
```

输出样例

```
18
```

考察知识点

gcd

难度系数 3

解题思路

这道题的解法在题目描述里以及很清楚的表述了。若 a,b 其中之一为 0，则最大公约数为非 0 的那个；若 a,b 均不为 0，则继续用此算法求解 b 和 a mod b 的 GCD，继续循环下去，直到其中一个为 0，输出不为 0 的那个数作为最大公约数。

代码

```
#include <stdio.h>

int gcd(int a, int b)
{
    return b ? gcd(b, a % b) : a;
```

```
}

int main()
{
	int a, b;
	while (scanf("%d%d", &a, &b) != EOF) {
		printf("%d\n", gcd(a, b));
	}
	return 0;
}
```