

2023 年常州市“程序设计小能手”比赛说明

- ◆ **文件保存的位置**：在不保护硬盘分区（由监考老师当场告知）
- ◆ **文件夹命名**：在不保护硬盘建立一个文件夹，文件夹的名称为选手的准考证号，例如：准考证号：128，文件夹名即为 128，本次比赛编写的程序必须存放在这个文件夹中。
- ◆ **文件名的要求**：在每题标题后的括号中有文件名，如：第一题 最大公约数（gcd.cpp），该题编好的程序保存时起名为 gcd。文件名错误不得分。
- ◆ 一般说来前面的题要比后面的容易，后面的题目虽然得到满分很难，然而拿一部分分数并不难。请合理分配你的时间，先保证程序的正确性，超时等问题都是次要的，计算机的运行速度往往比你想象的要快得多。如果某题不太会做你可以针对小数据编程争取拿部分分数，哪怕手算一个结果输出也行，比赛总是有难度的，不能像平时学校里的小测验那样老想着拿满分，从往年的经验来看你能得到总分的 1/3 就相当不错了。
- ◆ 所有测试点时限都是 1 秒，所有程序运行时内存都不能超过 256MB，大约可以存储六千万个 int 类型的整数。每题一般有 5 个以上测试点，并且测试点个数是该题分值的约数。如某题的满分为 100 分，则测试点通常会有 5,10,20,25 个。
- ◆ 输出时每行的行首不要有多余的空格，也不要有多余的空行，相邻两项输出之间严格用一个空格隔开，一行输出结束时一定要换行。评测时会忽略行末空格。
- ◆ 所有题目均使用标准输入输出，即从键盘输入数据，结果输出到屏幕，请认真阅读范例，你的程序请严格按范例程序的格式编写。
- ◆ 题目中用到的“^”符号表示乘方运算，如 $2^3=2 \times 2 \times 2=8$ ， $10^6=1000000$

【范例】

最大公约数和最小公倍数(gcdlcm.cpp)

问题描述

最大公约数(Greatest Common Divisor, 简称为 GCD): 如果有一个自然数 a 能被自然数 b 整除(也称 b 能整除 a , 记作 $b|a$), 则称 a 为 b 的倍数, b 为 a 的约数。两个自然数公共的约数, 叫做这两个自然数的公约数。所有公约数中最大的一个, 称为这两个自然数的最大公约数。最小公倍数(Least Common Multiple, 缩写 LCM): 对于两个自然数来说, 最小公倍数是指这两个数公共的倍数中最小的一个。例如: 在 12 和 16 中, 4 就是 12 和 16 的最大公约数。12 和 16 的最小公倍数是 48。

早在公元前 300 年左右，欧几里德就在他的著作《几何原本》中给出了求最小公倍数的高效方法——辗转相除法。辗转相除法使用到的原理很聪明也很简单，假设用 $\text{GCD}(x, y)$ 表示两个自然数 x 与 y 的最大公约数，取 $k = x / y$ ， $b = x \% y$ ，则 $x = k * y + b$ ，如果一个自然数能够同时整除 x 和 y ，则必能同时整除 b 和 y ；而能够同时整除 b 和 y 的自然数也必能同时整除 x 和 y ，即 x 和 y 的公约数与 b 和 y 的公约数是相同的，其最大公约数当然也相同，则当 $y \neq 0$ 时有 $\text{GCD}(x, y) = \text{GCD}(y, x \% y)$ ，如此便可把原问题转化为求两个更小的自然数的最大公约数，直到其中一个数为 0，剩下的另外一个数就是两者的最大公约数。以求 288 和 123 的最大公约数为例，操作如下： $288 \% 123 = 42$ $123 \% 42 = 39$ $42 \% 39 = 3$ $39 \% 3 = 0$ 所以 3 就是 288 和 123 的最大公约数。

计算最小公倍数时，通常会借助最大公约数来辅助计算。可以证明两个自然数的乘积等于它们的最大公约数和最小公倍数的乘积，即 $a \times b = \text{GCD}(a, b) \times \text{LCM}(a, b)$ 。如 $12 \times 16 = 192 = \text{GCD}(12, 16) \times \text{LCM}(12, 16) = 4 \times 48$ 。

编一个程序对于输入的两个自然数 a 和 b ，求它们的最大公约数和最小公倍数。

输入格式

输入数据仅有一行包含两个用空格隔开的自然数 a 和 b ，范围不超过 int 。

输出格式

输出数据仅有一行包含两个自然数，表示要求的最大公约数和最小公倍数。两数之间严格用一个空格隔开，行末没有多余的空格。

样例输入

12 16

样例输出

4 48

以下是 C++ 源程序，存盘文件名为 `gcdlcm.cpp`

```
#include <bits/stdc++.h>
using namespace std;
int main(){
    long long m,n,a,b,r;
    cin>>m>>n;
    a = m;
    b = n;
    while ( b != 0 ){
        r = a % b;
        a = b;
        b = r;
    }
    cout<<a<<" "<<m*n/a<<endl;
    return 0;
}
```

注意敏感词不能在程序中使用，如 `y1,y2,y3,left,right,mid,link,next` 等！

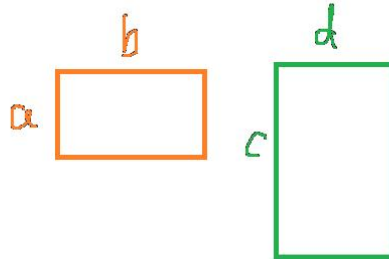
矩形纸片(rectangle.cpp)

本题总分：60 分

问题描述

小 X 有一个 10^9 行 10^9 列的网格纸，每个格子都是一个单位长度的正方形。

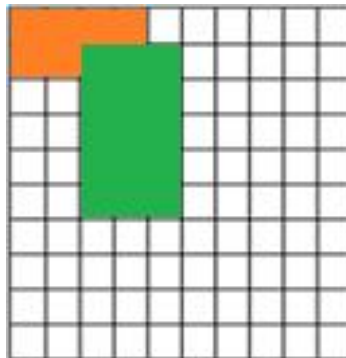
现在小 X 还有两个矩形纸片，矩形的边长都是单位长度的整数倍。第一个矩形纸片的长为 a ，宽为 b ，第二个矩形纸片的长为 c ，宽为 d 。



小 X 把第一个纸片放在了整张网格纸的左上角，再把第二个纸片的左上角放在第 x 行第 y 列的格子的左上角上，并且不旋转或者翻转纸片。

小 X 想知道两张纸片覆盖网格纸的总面积。

例如，当 $a=2, b=4, c=5, d=3, x=2, y=3$ 时，覆盖情况如下图所示，覆盖的总面积为 21 个单位面积。



输入格式

一行 6 个正整数 a, b, c, d, x, y 。

输出格式

输出两张纸片覆盖网格纸的总面积。

样例输入 1

2 4 5 3 2 3

样例输出 1

21

样例输入 2

4 5 2 8 1 1

样例输出 2

26

样例输入 3

5 5 5 5 2 3

样例输出 3

38

数据范围

本题共有 6 个测试点，每个测试点 10 分。

对于所有测试点： $1 \leq a, b, c, d, x, y \leq 1000$ 。

对于测试点 1-2：保证 $a=b=c=d$ 。

对于测试点 3-4：保证 $x=y=1$ 。

奶牛农场(cow.cpp)

本题总分：90 分

问题描述

小 X 是 CZ 市著名的农场主，他拥有着 CZ 市最大的奶牛农场。农场里有一排牛棚，一共 n 个牛棚，从左到右依次编号为 $1, 2, \dots, n$ 。目前有些牛棚里住着奶牛，有些牛棚还是空的。每个奶牛有一个高度，其中第 i 个牛棚里的奶牛的高度为 $H[i]$ ，如果第 i 个牛棚里没有奶牛的话则 $H[i]=0$ ，为了使小 X 的牛棚变得美观，他打算去市场上买一些奶牛放到空着的牛棚里（假设市场上能买到任意多头高度在 1 到 10^9 之间的任意正整数的奶牛），使得每个牛棚里都有一头奶牛，并且高度从左往右严格递增。

请你告诉小 X 是否能让他的牛棚变得美观，如果可以请给出一个任意合法的方案。

输入格式

第一行 1 个正整数 n ，表示牛棚个数。

第二行 n 个非负整数 $H[i]$ ，如果 $H[i]=0$ 说明第 i 个牛棚是空的，否则说明第 i 个牛棚里面有一头高度为 $H[i]$ 的奶牛。

输出格式

第一行输出一个字符串“YES”或“NO”。如果能让他的牛棚变得美观，则输出“YES”，否则输出“NO”。（均不包含引号）

如果第一行输出“YES”，再输出第二行 n 个正整数 $1 \leq H'[i] \leq 10^9$ ，你需要保证对所有 $1 \leq i \leq n-1$ 满足 $H'[i] < H'[i+1]$ ，并且如果 $H[i] > 0$ ，那么 $H'[i] = H[i]$ ，如果有多种合法的方案，输出任意一种即可。

样例输入 1

```
3
0 0 0
```

样例输出 1

```
YES
4 5 6
```

样例输入 2

```
4
0 2 0 4
```

样例输出 2

```
YES
1 2 3 4
```

样例输入 3

```
4
0 0 0 2
```

样例输出 3

```
NO
```

样例 3 解释

因为高度是正整数，还要严格递增，所以第 4 头奶牛的高度必须 ≥ 4 。所以不存在满足题目条件的方案。

样例输入 4

2
1000000000 0

样例输出 3

NO

样例 3 解释

因为买不到高度大于 10^9 的奶牛，所以不存在满足题目条件的方案。

数据范围

本题共有 9 个测试点，每个测试点 10 分。

对于测试点 1-5： $1 \leq n \leq 5, 0 \leq H[i] \leq 10$ 。

对于测试点 6-9： $1 \leq n \leq 10^5, 0 \leq H[i] \leq 10^9$ 。

ABC 字符串 (string.cpp)

本题总分：100 分

问题描述

小 Y 给了小 X 一个只包含大写字母 A,B,C 的字符串。你可以对这个字符串进行如下操作：将子串 ABC 变成 BCA。

小 X 想知道这个字符串最多能进行多少次操作。

一个字符串的子串是把这个字符串通过删去头部和尾部若干个字符形成的字符串。

例如：A,B,BB,AB,ABB 是 ABB 的子串，ABA 不是 ABBA 的子串。

输入格式

一行一个字符串 S。

输出格式

一行一个整数表示答案。

样例输入 1

ABCABC

样例输出 1

3

样例解释 1

ABCABC

ABCBCA

BCABCA

BCBCAA

最多操作 3 次。

样例输入 2

ABCACCBABCBCAABCB

样例输出 2

6

数据范围

本题共有 10 个测试点，每个测试点 10 分。

对于全部测试点： $n \leq 200000$ ，n 表示输入的字符串的长度。

对于测试点 1-4： $n \leq 10$

对于测试点 5-7： $n \leq 1000$ ，并且保证无论按照什么顺序操作，被操作的子串两两不相交（换句话说，一个下标不会被两个操作的字符串同时覆盖）

对于测试点 8-10： $n \leq 200000$

数学作业 (fib.cpp)

本题总分：100 分

问题描述

今天小 X 的数学老师带领大家学习了斐波那契序列：

斐波那契序列指的是这样一个数列：1、2、3、5、8、13、21、34。从第 3 个数开始，每个数都是前两个数的和，比如 $8=3+5$ ， $34=13+21$ 。数列里的数叫做斐波那契数。

一个数 n 的斐波那契表示是指把 n 写成若干个互不相同的斐波那契数的和。一个数可以有多种不同的斐波那契表示。比如 14 有三种斐波那契表示： $14=1+13$ ， $14=1+5+8$ ， $14=1+2+3+8$ 。

数学老师给小 X 留下了一个数学作业，她告诉小 X 一个正整数 n ，想让小 X 算出 n 有多少种不同的斐波那契表示。

小 X 请你帮助他完成他的数学作业。

输入格式

第一行为一个正整数 n 。

输出格式

输出一行一个整数表示答案。

样例输入 1

14

样例输出 1

3

样例解释 1

见题面。

样例输入 2

1110

样例输出 2

21

样例输入 3

1000000000000

样例输出 3

283392

数据范围

本题共有 10 个测试点，每个测试点 10 分。

对于测试点 1-5： $1 \leq n \leq 10^4$

对于测试点 6-8： $1 \leq n \leq 10^9$

对于测试点 9-10： $1 \leq n \leq 10^{12}$

洗牌(card.cpp)

本题总分：110 分

问题描述

小 X 有 n 张标有数字的纸牌，第 i 张纸牌上面的数是 $a[i]$ ，现在小 X 想通过洗牌打乱它们的顺序。

对于一个洗牌后的顺序，小 X 觉得相邻两张纸牌上数的差的绝对值之和越大，牌就洗的越乱。

举个例子：现在有 4 张纸牌，纸牌上的数依次为 1,2,3,4。

假设洗完牌后，纸牌上的数依次 4,3,2,1，相邻两张纸牌上数的差的绝对值之和为 $|4-3|+|3-2|+|2-1|=1+1+1=3$ 。

假设洗完牌后，纸牌上的数依次 2,4,1,3，相邻两张纸牌上数的差的绝对值之和为 $|4-2|+|4-1|+|3-1|=2+3+2=7$ 。

那么小 X 就会觉得 2,4,1,3 的顺序比 4,3,2,1 更乱。

小 X 想要问问你，对于所有顺序，相邻两张纸牌上数的差的绝对值之和最大能是多少。

输入格式

第一行 1 个正整数 n ，表示纸牌的张数。

第二行 n 个正整数 $a[i]$ ，表示第 i 张纸牌上的数。注意具有相同数字的纸牌可能有多张。

输出格式

输出一行一个整数，表示答案。

样例输入 1

```
4
1 2 3 4
```

样例输出 1

```
7
```

样例输入 2

```
5
1 2 3 4 5
```

样例输出 2

```
11
```

样例解释 2

一种可行的顺序是 3 5 1 4 2

样例输入 3

```
10
1 2 3 4 5 6 7 8 9 10
```

样例输出 3

49

数据范围

本题共有 11 个测试点，每个测试点 10 分。

保证当测试点编号是偶数时， n 也是偶数。

对于测试点 1-3: $1 \leq n \leq 10, 1 \leq a[i] \leq 1000000$

对于测试点 4-8: $1 \leq n \leq 100, 1 \leq a[i] \leq 10$

对于测试点 9-11: $1 \leq n \leq 100000, 1 \leq a[i] \leq 1000000$

红绿灯(light.cpp)

本题总分：120 分

问题描述

小 X 家门前有两个红绿灯，小 X 做完了数学作业，闲着无聊便在窗边观察。他发现这两个红绿灯亮红灯和亮绿灯的时间是相等的，第一个红绿灯亮 p 秒绿灯，再亮 p 秒红灯.....，第二个红绿灯亮 q 秒绿灯，再亮 q 秒红灯.....，如此循环往复。

现在恰好两个红绿灯都从红灯变成了绿灯，小 X 想要知道未来的 $2pq$ 秒内，有多少秒满足两个红绿灯都亮绿灯。

输入格式

第一行 2 个正整数 p, q ，含义见题面。

输出格式

输出一行一个整数表示在未来的 $2pq$ 秒内，有多少秒满足两个红绿灯都亮绿灯。

样例输入 1

2 3

样例输出 1

3

样例解释 1

在未来的 12 秒内，第一个红绿灯在第 1, 2, 5, 6, 9, 10 秒亮绿灯。

第一个红绿灯在第 1, 2, 3, 7, 8, 9 秒亮绿灯。

在第 1, 2, 9 秒时，同时亮绿灯，一共 3 秒。

样例输入 2

18 66

样例输出 2

612

样例输入 3

1 255

样例输出 3

128

数据规模

本题共有 12 个测试点，每个测试点 10 分。

对于测试点 1-3： $1 \leq p, q \leq 1000$

对于测试点 4-5： $p=1, 1 \leq q \leq 10^9$

对于测试点 6-9： $1 \leq p, q \leq 10^9$ 且 p, q 互质，即 p, q 的最大公约数是 1

对于测试点 10-12： $1 \leq p, q \leq 10^9$

积木(block.cpp)

本题总分：120 分

问题描述

小 X 在地上玩积木，每块积木都是一个 $1*1*1$ 的正方体。地面可以看成一个 $n*m$ 的网格，其中每一小格内都整齐地从下到上堆着若干块积木。其中第 i 行第 j 列中有 $h[i][j]$ 块积木。

现在小 X 想要拿走一些积木，使得剩下来到积木组成一个正方体，正方体指的是长宽高都相同的长方体。

小 X 想问你他最少拿掉多少块积木才能使得最后剩下的积木组成一个正方体。

输入格式

第一行，2 个整数 n 和 m 表示地面的大小。

接下来 n 行，每行 m 个非负整数。第 i 行第 j 个数表示 $h[i][j]$ 。

输出格式

一行一个整数表示答案。

样例输入 1

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

样例输出 1

```
10
```

样例解释 1

拿完之后每个位置的积木数为：

```
2 2 0
2 2 0
0 0 0
```

一共拿掉 $(2-2)+(2-2)+(1-0)+(3-2)+(2-2)+(2-0)+(3-0)+(1-0)+(2-0)=1+1+2+3+1+2=10$ 块积木。

样例输入 2

```
5 5
4 4 3 4 3
3 4 3 3 3
3 3 1 4 4
3 4 4 3 3
4 3 4 4 4
```

样例输出 2

```
77
```

数据范围

本题共有 12 个测试点，每个测试点 10 分。

对于所有测试点 $1 \leq n, m \leq 1000, 0 \leq h[i][j] \leq 1000$

对于测试点 1-3 : $1 \leq n, m \leq 50$

对于测试点 4-6 : $1 \leq n, m \leq 200$

对于测试点 7-9 : $1 \leq n, m \leq 1000, 0 \leq h[i][j] \leq 20$

