第六章课后习题：

### 题目1 : 博弈游戏·Nim游戏

时间限制:10000ms

单点时限:1000ms

内存限制:256MB

|  |
| --- |
|  |

#### **描述**

今天我们要认识一对新朋友，Alice与Bob。  
Alice与Bob总是在进行各种各样的比试，今天他们在玩一个取石子的游戏。  
在这个游戏中，Alice和Bob放置了N堆不同的石子，编号1..N，第i堆中有A[i]个石子。  
每一次行动，Alice和Bob可以选择从一堆石子中取出任意数量的石子。至少取1颗，至多取出这一堆剩下的所有石子。  
Alice和Bob轮流行动，取走最后一个石子的人获得胜利。  
假设每一轮游戏都是Alice先行动，请你判断在给定的情况下，如果双方都足够聪明，谁会获得胜利？

[提示：Nim?!](https://hihocoder.com/contest/lanqiaoalgohw6/problem/1)

#### **输入**

第1行：1个整数N。表示石子堆数。1≤N≤100  
第2行：N个整数，第i个整数表示第i堆石子的个数A[i]，1≤A[i]≤10000

#### **输出**

第1行：1个字符串，若Alice能够获胜输出"Alice"，否则输出"Bob"

**样例输入**

3

3 2 1

**样例输出**

Bob

### 题目2 : 数论四·扩展欧几里德

时间限制:10000ms

单点时限:1000ms

内存限制:256MB

|  |
| --- |
|  |

#### **描述**

小Hi和小Ho周末在公园溜达。公园有一堆围成环形的石板，小Hi和小Ho分别站在不同的石板上。已知石板总共有m块，编号为 0..m-1，小Hi一开始站在s1号石板上，小Ho一开始站在s2号石板上。

小Hi：小Ho，你说我们俩如果从现在开始按照固定的间隔数同时同向移动，我们会不会在某个时间点站在同一块石板上呢？

小Ho：我觉得可能吧，你每次移动v1块，我移动v2块，我们看能不能遇上好了。

小Hi：好啊，那我们试试呗。

一个小时过去了，然而小Hi和小Ho还是没有一次站在同一块石板上。

小Ho：不行了，这样走下去不知道什么时候才汇合。小Hi，你有什么办法算算具体要多久才能汇合么？

小Hi：让我想想啊。。

[提示：扩展欧几里德](https://hihocoder.com/contest/lanqiaoalgohw6/problem/2)

#### **输入**

第1行：每行5个整数s1,s2,v1,v2,m，0≤v1,v2≤m≤1,000,000,000。0≤s1,s2<m

中间过程可能很大，最好使用64位整型

#### **输出**

第1行：每行1个整数，表示解，若该组数据无解则输出-1

**样例输入**

0 1 1 2 6

**样例输出**

5

### 题目3 : 数论六·模线性方程组

时间限制:10000ms

单点时限:1000ms

内存限制:256MB

|  |
| --- |
|  |

#### **描述**

小Ho：今天我听到一个挺有意思的故事！

小Hi：什么故事啊？

小Ho：说秦末，刘邦的将军韩信带领1500名士兵经历了一场战斗，战死四百余人。韩信为了清点人数让士兵站成三人一排，多出来两人；站成五人一排，多出来四人；站成七人一排，多出来六人。韩信立刻就知道了剩余人数为1049人。

小Hi：韩信点兵嘛，这个故事很有名的。

小Ho：我觉得这里面一定有什么巧妙的计算方法！不然韩信不可能这么快计算出来。

小Hi：那我们不妨将这个故事的数学模型提取出来看看？

小Ho：好！

<小Ho稍微思考了一下>

小Ho：韩信是为了计算的是士兵的人数，那么我们设这个人数为x。三人成排，五人成排，七人成排，即x mod 3, x mod 5, x mod 7。也就是说我们可以列出一组方程：

x mod 3 = 2

x mod 5 = 4

x mod 7 = 6

韩信就是根据这个方程组，解出了x的值。

小Hi：嗯，就是这样！我们将这个方程组推广到一般形式：**给定了n组除数m[i]和余数r[i]，通过这n组(m[i],r[i])求解一个x，使得x mod m[i] = r[i]。**

小Ho：我怎么感觉这个方程组有固定的解法？

小Hi：这个方程组被称为**模线性方程组**。它确实有固定的解决方法。不过在我告诉你解法之前，你不如先自己想想怎么求解如何？

小Ho：好啊，让我先试试啊！

[提示：模线性方程组](https://hihocoder.com/contest/lanqiaoalgohw6/problem/3)

#### **输入**

第1行：1个正整数, N，2≤N≤1,000。

第2..N+1行：2个正整数, 第i+1行表示第i组m,r，2≤m≤20,000,000，0≤r<m。

计算过程中尽量使用64位整型。

#### **输出**

第1行：1个整数，表示满足要求的最小X，若无解输出-1。答案范围在64位整型内。

**样例输入**

3

3 2

5 3

7 2

**样例输出**

23

### 题目4 : 数论二·Eular质数筛法

时间限制:10000ms

单点时限:1000ms

内存限制:256MB

|  |
| --- |
|  |

#### **描述**

小Ho：小Hi，上次我学会了如何检测一个数是否是质数。于是我又有了一个新的问题，我如何去快速得求解[1,N]这个区间内素数的个数呢？

小Hi：你自己有什么想法么？

小Ho：有！我一开始的想法是，自然我们已经知道了如何快速判定一个数是否是质数，那么我就直接将[1,N]之间每一个数判定一次，就可以得到结果。但我发现这个方法太笨了。

小Hi：确实呢，虽然我们已经通过快速素数检测将每一次判定的时间复杂度降低，但是N个数字的话，总的时间复杂度依旧很高。

小Ho：是的，所以后来我改变了我的算法。我发现如果一个数p是质数的话，那么它的倍数一定都是质数。所以我建立了一个布尔类型的数组isPrime，初始化都为true。我从2开始枚举，当我找到一个isPrime[p]仍然为true时，可以确定p一定是一个质数。接着我再将N以内所有p的倍数全部设定为isPrime[p\*i]=false。

写成伪代码为：

isPrime[] = true

primeCount = 0

For i = 2 .. N

If isPrime[i] Then

primeCount = primeCount + 1

multiple = 2

While (i \* multiple ≤ N)

isPrime[i \* multiple] = false

multiple = multiple + 1

End While

End If

End For

小Hi：小Ho你用的这个算法叫做Eratosthenes筛法，是一种非常古老的质数筛选算法。其时间复杂度为O(n log log n)。但是这个算法有一个冗余的地方：比如合数10，在枚举2的时候我们判定了一次，在枚举5的时候我们又判定了一次。因此使得其时间复杂度比O(n)要高。

小Ho：那有没有什么办法可以避免啊？

小Hi：当然有了，一个改进的方法叫做Eular筛法，其时间复杂度是O(n)的。

[提示：Eular质数筛法](https://hihocoder.com/contest/lanqiaoalgohw6/problem/4)

#### **输入**

第1行：1个正整数n，表示数字的个数，2≤n≤1,000,000。

#### **输出**

第1行：1个整数，表示从1到n中质数的个数

**样例输入**

9

**样例输出**

4