

Bài 0. Danh sách ước của một số

Divisor.cpp

Cho: Một danh sách các ước số của một số tự nhiên x , ngoại trừ 1 và chính x .

Yêu cầu:

- Tìm số tự nhiên x nhỏ nhất có thể từ danh sách các ước số đã cho.
- Nếu không tồn tại số x thỏa mãn, in ra -1.

Điều kiện:

- Số lượng ước số trong danh sách không quá 300.
- Các ước số trong danh sách đều khác nhau và lớn hơn hoặc bằng 2.

Đầu vào:

- Số lượng bộ test t ($1 \leq t \leq 25$).
- Với mỗi test case:
 - Số lượng ước số n ($1 \leq n \leq 300$).
 - Danh sách n ước số của x : d_1, d_2, \dots, d_n ($2 \leq d_i \leq 10^6$). Các số là khác nhau.

Đầu ra:

- Với mỗi test case, in ra số x tìm được hoặc -1 nếu không tồn tại.

Ví dụ:

Input	Output	Giải thích
3	48	
8	4	
8 2 12 6 4 24 16 3	-1	
1		
2		
4		
2 3 4 5		

Bài 1. Trò chơi chia số

DIVGAME.cpp

Cho số nguyên dương N . Xét việc lặp lại thao tác sau trên N :

- Đầu tiên, chọn một số nguyên dương z thỏa mãn tất cả các điều kiện sau:
 - z có luôn được biểu diễn dưới dạng $z = p^e$, trong đó p là số nguyên tố và e là số nguyên dương.
 - z là ước của N .
 - z khác với tất cả các số nguyên đã chọn trong các thao tác trước đó.
- Sau đó, thay thế N bằng N/z .

Tìm số lần tối đa có thể áp dụng thao tác.

Ràng buộc

- Tất cả giá trị đầu vào là số nguyên.
- N nằm trong khoảng từ 1 đến 10^{14} (bao gồm cả 1 và 10^{14}).

Đầu vào

- Đầu vào được nhập từ giao diện chuẩn (Standard Input) theo định dạng sau:
 - Một dòng duy nhất chứa số nguyên N .

Đầu ra

- In ra số lần tối đa có thể áp dụng phép toán.

Bài 2. Số có 9 ước

CHINUOC.cpp

Cho một số nguyên dương N , hãy viết chương trình đếm số lượng số có đúng 9 ước và không lớn hơn N .

INPUT: Một dòng duy nhất ghi số nguyên dương N ($0 < N \leq 10^9$).

OUTPUT: In ra kết quả bài toán.

Ví dụ

• input
1000
output
8

Bài toán : Biến đổi về bằng 1

One.cpp

Bạn có một dãy số nguyên a có độ dài n . Bạn có thể thực hiện các phép toán. Mỗi phép toán như sau: chọn hai số nguyên liên tiếp từ a , gọi là x và y , và thay thế một trong hai số đó bằng $\gcd(x, y)$, trong đó \gcd là ước số chung lớn nhất.

Bạn cần tìm số phép toán tối thiểu để biến tất cả các số thành 1.

Dữ liệu vào:

Dòng đầu tiên chứa một số nguyên n ($1 \leq n \leq 2000$) - độ dài của dãy số. Dòng thứ hai chứa n số nguyên a_1, a_2, \dots, a_n ($1 \leq a_i \leq 10^9$) - các phần tử của dãy số.

Dữ liệu ra:

In ra -1 nếu không thể biến tất cả các số thành 1. Nếu có thể, in ra số phép toán tối thiểu cần thiết để biến tất cả các số thành 1.

Ví dụ:

Input	Output	Giải thích
5 2 2 3 4 6	5	
4 2 4 6 8	-1	
3 2 6 9	4	

Bài 3. Tìm ước chung lớn nhất

LCM.cpp

Cho một tập hợp các số nguyên dương $s = \{s_1, s_2, \dots, s_k\}$. Định nghĩa ước chung lớn nhất (GCD) và bội chung nhỏ nhất (LCM) của s như sau:

- $\gcd(s)$ là số nguyên dương lớn nhất x sao cho tất cả các số nguyên trong s đều chia hết cho x .
- $\text{lcm}(s)$ là số nguyên dương nhỏ nhất x sao cho x chia hết cho tất cả các số nguyên trong s .

Ví dụ: $\gcd(\{8, 12\}) = 4$, $\gcd(\{12, 18, 6\}) = 6$ và $\text{lcm}(\{4, 6\}) = 12$. Lưu ý rằng với bất kỳ số nguyên dương x nào, $\gcd(\{x\}) = \text{lcm}(\{x\}) = x$.

Orac có một dãy số a có độ dài n . Anh ấy tạo ra một tập hợp các số $t = \{\text{lcm}(\{a_i, a_j\}) \mid i < j\}$ và yêu cầu bạn tìm giá trị của $\gcd(t)$. Nói cách khác, bạn cần tính ước chung lớn nhất của các bội chung nhỏ nhất của tất cả các cặp phần tử trong dãy đã cho.

Đầu vào

- Dòng đầu tiên chứa một số nguyên n ($2 \leq n \leq 100000$).

- Dòng thứ hai chứa n số nguyên a_1, a_2, \dots, a_n ($1 \leq a_i \leq 200000$).

Đầu ra

- In ra một số nguyên: $\gcd(\{\text{lcm}(\{a_i, a_j\}) \mid i < j\})$.

Ví dụ

Input	Output	Giải thích
2 1 1	1	Ví dụ 1: Vì chỉ có một cặp số là (1, 1) nên tập hợp t chỉ có một phần tử là $\text{lcm}(1, 1) = 1$. Do đó, $\gcd(t) = 1$.
4 10 24 40 80	40	Ví dụ 2: Tập hợp t chứa các giá trị lcm của tất cả các cặp số trong dãy. Sau khi tính toán, ta được tập hợp $t = \{120, 40, 80, 120, 240, 80\}$. Ước chung lớn nhất của các số trong tập hợp này là 40.
10 540 648 810 648 720 540 594 864 972 648	54	

Bài tập (khó): Cho số nguyên dương N, tìm dãy các số nguyên dương $a_1 \leq a_2 \leq a_3 \leq \dots \leq a_M$ sao cho: $a_1 + a_2 + \dots + a_M = N$ và $\text{LCM}(a_1, a_2, \dots, a_M)$ đạt giá trị lớn nhất. Khi đó hãy in ra dãy a_1, a_2, \dots, a_M tương ứng.

Ví dụ

Input	Output
5	2 3
14	3 4 7

Giới hạn: $N \leq 10^4$.

Bài 4. Phép chia dư (495B)

Modular.cpp

- Tìm số nghiệm của phương trình đồng dư $a \% x = b$ với a và b là các số nguyên không âm.

Đầu vào:

- Hai số nguyên không âm a và b cách nhau một khoảng trắng. $0 \leq a, b \leq 10^9$.

Đầu ra:

- Nếu có vô số nghiệm, in ra "infinity".
- Ngược lại, in ra số lượng nghiệm của phương trình.

Ví dụ:

Input	Output	Giải thích
21 5	2	Có 2 số x là 8 và 16 thỏa mãn: $21 \% 8 = 5$ và $21 \% 16 = 5$.
9435152 272	282	
10 10	infinity	Có vô số số x thỏa mãn

Bài 5. Tổng ước sumdiv.cpp

Cho hai số tự nhiên A và B. Gọi S là tổng của tất cả các ước tự nhiên của A^B . Tìm S modulo 1.000.000.007 (số dư khi chia S cho 1.000.000.007).

Dữ liệu vào:

- Dòng đầu tiên của file sumdiv.in chứa hai số A và B, cách nhau ít nhất một dấu cách.

Dữ liệu ra:

- Dòng đầu tiên của file sumdiv.out chứa giá trị S modulo 1.000.000.007.

Điều kiện:

- $1 \leq A \leq 10^{12}$
- $0 \leq B \leq 10^{18}$

Giải thích:

- A^B là một lũy thừa với cơ số A được nâng lên lũy thừa B.
- Ước tự nhiên của một số là những số chia hết cho số đó.
- Tổng S là tổng của tất cả các ước tự nhiên của A^B .
- Cần tìm số dư khi chia S cho 1.000.000.007.

Bài 6. Chuỗi số may mắn (1766D)**lucky.cpp**

<https://codeforces.com/blog/entry/110066>

- Cho một tập hợp gồm n cặp số nguyên dương (x, y).
- Một cặp số (x, y) được gọi là may mắn nếu ước chung lớn nhất của x và y bằng 1 ($\gcd(x, y) = 1$).
- Một chuỗi được tạo bởi các cặp số liên tiếp (x, y), (x+1, y+1), (x+2, y+2), ..., (x+k, y+k) với $k \geq 0$.
- Một chuỗi may mắn là một chuỗi mà tất cả các cặp số trong chuỗi đều là may mắn.
- Tìm độ dài chuỗi may mắn dài nhất cho mỗi cặp số trong tập hợp đã cho. Nếu không tồn tại chuỗi may mắn, in ra -1.

Đầu vào

- Dòng đầu tiên chứa số nguyên n ($1 \leq n \leq 10^6$) là số lượng cặp số.
- n dòng tiếp theo, mỗi dòng chứa hai số nguyên x và y ($1 \leq x < y \leq 10^7$) là một cặp số.

Đầu ra

- In ra n số nguyên, mỗi số là độ dài chuỗi may mắn dài nhất cho cặp số tương ứng. Nếu không tồn tại chuỗi may mắn dài vô hạn, in ra -1.

Ví dụ:

Input	Output	Giải thích
4	0	
5 15	1	
13 37	-1	
8 9	79	
10009 20000		

Ví dụ 1: Cặp số (5, 15) không phải là số may mắn vì ước chung lớn nhất của 5 và 15 là 5, không bằng 1.

Do đó, chuỗi may mắn bắt đầu từ cặp số (5, 15) có độ dài là 0.

Ví dụ 2: Cặp số (13, 37) là số may mắn vì ước chung lớn nhất của 13 và 37 là 1.

Tuy nhiên, cặp số tiếp theo (14, 38) không phải là số may mắn vì ước chung lớn nhất của 14 và 38 là 2, không bằng 1.

Do đó, chuỗi may mắn bắt đầu từ cặp số (13, 37) chỉ có độ dài 1, bao gồm duy nhất cặp số (13, 37).

Bài 7. Hàm kì lạ Function.cpp

Định nghĩa $f(i)$ là số nguyên dương x bé nhất mà x không là ước của i .

Hãy tính $S = \sum_{i=1}^n f(i)$ với $1 \leq n \leq 10^{14}$. Hay nói cách khác là với mỗi số

n , hãy tính tổng: $S = f(1) + f(2) + f(3) + \dots + f(n)$. Do kết quả có thể quá lớn, nên hãy in ra kết quả của tổng S đó.

Đầu vào: Ghi một số tự nhiên n duy nhất.

Đầu ra: Ghi ra giá trị của S .

Ví dụ:

Input	Output	Giải thích
4	10	Xem bên dưới
10	26	
10000000000000000	366580019	

Giải thích ví dụ 1: với $n = 4$

$f(1) = 2$ vì 2 là số nguyên dương nhỏ nhất không phải ước của 1.

$f(2) = 3$ vì 3 là số nguyên dương nhỏ nhất không phải ước của 2.

$f(3) = 2$ vì 2 là số nguyên dương nhỏ nhất không phải ước của 3.

$f(4) = 3$ vì 3 là số nguyên dương nhỏ nhất không phải ước của 4.

Do vậy tổng $f(1) + f(2) + f(3) + f(4) = 2 + 3 + 2 + 3 = 10$.

Subtask 1: 30% test có $n \leq 10^4$

Subtask 2: 40% test có $n \leq 10^6$;

Subtask 3: 430% test có $n \leq 10^{14}$.

Bài 8.

Cho hai số nguyên a, m ($1 \leq a < m < 10^{10}$). Hãy đếm số lượng số x thỏa mãn $0 \leq x < m$ và $GCD(a, m) = GCD(a + x, m)$.

Dữ liệu vào:

- Số lượng test case T ($1 \leq T \leq 50$).
- Với mỗi test case, hai số nguyên a và m ($1 \leq a < m \leq 10^{10}$).

Dữ liệu ra:

- Với mỗi test case, in ra số lượng giá trị x thỏa mãn điều kiện.

Yêu cầu:

- Viết chương trình tính toán và in ra kết quả cho từng test case.

Ví dụ:

Input	Output	Giải thích
3	6	
4 9	1	
5 10	9999999966	
42 9999999967		

Ví dụ 1:

- $a = 4, m = 9$
- Các giá trị x thỏa mãn là: 0, 1, 3, 4, 6, 7.
- Trong mỗi trường hợp, $\gcd(4, 9) = \gcd(4+x, 9) = 1$.

Ví dụ 2:

- $a = 5, m = 10$
- Chỉ có giá trị $x = 0$ thỏa mãn, vì $\gcd(5, 10) = \gcd(5+0, 10) = 5$.

Ví dụ 3:

- $a = 42, m = 9999999967$
- Do m là một số nguyên tố lớn, nên hầu hết các giá trị x từ 0 đến $m-1$ sẽ thỏa mãn điều kiện.
- Kết quả là 9999999966.

Bài 9. Chia dãy số**Partition.cpp**

Cho một dãy số nguyên a_1, a_2, \dots, a_n . Với mỗi ước số dương k là ước của n , Allen thực hiện các bước sau:

1. Chia dãy thành n/k đoạn con không giao nhau, mỗi đoạn có độ dài k .
2. Nếu tồn tại số nguyên dương m (lớn hơn hoặc bằng 2) sao cho khi thay thế mỗi phần tử trong dãy bằng số dư của nó khi chia cho m , thì tất cả các đoạn con trở nên giống nhau, Allen sẽ được 1 điểm.

Yêu cầu tìm tổng số điểm mà Allen có thể đạt được.

Input

- Dòng đầu tiên chứa số nguyên t ($1 \leq t \leq 10^4$) - số lượng bộ test.
- Mỗi bộ test gồm hai dòng:
 - Dòng đầu tiên chứa số nguyên n ($1 \leq n \leq 2 \cdot 10^5$) - độ dài của dãy a .
 - Dòng thứ hai chứa n số nguyên a_1, a_2, \dots, a_n ($1 \leq a_i \leq n$) - các phần tử của dãy a .
- Tổng của n trên tất cả các bộ test không vượt quá $2 \cdot 10^5$.

Output

- Với mỗi bộ test, in ra một số nguyên duy nhất - số điểm mà Allen đạt được.

Chú ý:

- n là độ dài của dãy số.
- k là một ước số của n .
- m là một số nguyên dương lớn hơn hoặc bằng 2.
- Hai đoạn con được coi là giống nhau nếu các phần tử tương ứng của chúng bằng nhau sau khi chia lấy dư cho m .

Ví dụ:

Input	Output	Giải thích
7	2	Test thứ nhất có thể có 2 cách chia là: 1,2/1,4
4	1	chọn $k=2, m=2$ thu được dãy 1,0/1,0. Chọn
1 2 1 4	2	$k=1$ được dãy 1,2,1,4
3	4	Test cuối cùng có thể chọn $k=2$, ta có 2 đoạn
1 2 3	4	1,7,5,1,4/3,1,3,1,4 chọn $m=2$ ta được
5	1	1,1,1,1,0/1,1,1,1,0

1 1 1 1 1 6 1 3 1 1 3 1 6 6 2 6 2 2 2 6 2 6 3 6 6 6 10 1 7 5 1 4 3 1 3 1 4	2	
--	---	--

Bài 10. Thêm điểm

Point.cpp

Trên một đường thẳng có n điểm, điểm thứ i có tọa độ là x_i . Tất cả các tọa độ đều khác nhau. Hãy xác định số lượng điểm m nhỏ nhất cần thêm vào đường thẳng để khoảng cách giữa mọi cặp điểm liên tiếp trở nên bằng nhau.

Input:

- Dòng đầu tiên chứa một số nguyên duy nhất n ($3 \leq n \leq 100000$) - số lượng điểm.
- Dòng thứ hai chứa một dãy các số nguyên x_1, x_2, \dots, x_n ($-10^9 \leq x_i \leq 10^9$) - tọa độ của các điểm. Tất cả các tọa độ đều khác nhau. Các điểm có thể được đưa ra theo thứ tự bất kỳ.

Output:

- In ra một số nguyên duy nhất m - số lượng điểm cần thêm vào để khoảng cách giữa mọi cặp điểm liên tiếp trở nên bằng nhau.

Ví dụ:

Input	Output	Giải thích
3 -5 10 5	1	Thêm một điểm ở tọa độ 0
6 100 200 400 300 600 500	0	Không cần thêm điểm nào
4 10 9 0 -1	8	Thêm để tạo các điểm liên tiếp

Bài 11. Phân tích số

Factorization.cpp

Cho một số nguyên dương N , ta luôn có thể phân tích N ra thành tích các lũy thừa như sau: $N = a_1^{p_1} \cdot a_2^{p_2} \dots a_k^{p_k}$, ($1 < a_i$). Trong đó các số a_i chỉ có thể phân tích thành **tích các thừa số nguyên tố khác nhau**.

Ví dụ, ta có $N = 28$ thì cách phân tích $N = 2^2 \cdot 7^1$ là hợp lệ, còn cách phân tích $N = 4^1 \cdot 7^1$ là không hợp lệ vì có $4 = 2^2$.

Yêu cầu: Với mỗi số N cho trước, hãy tìm giá trị lớn nhất có thể của:

$$S(N) = a_1 \cdot p_1 + a_2 \cdot p_2 + \dots + a_k \cdot p_k$$

Input: Dòng đầu ghi số T là số testcase, mỗi dòng sau ghi một số N tương ứng. $T \leq 1000, N \leq 10^9$.

Output: In ra kết quả lớn nhất của $S(N)$ với mỗi testcase tương ứng.

Ví dụ:

Input	Output	Giải thích
7	20	Trong ví dụ $N=100$ thì phân tích $100 = 10^2$ có $S(100) = 20$ là lớn nhất. Với $N=10$ thì $10 = 10^1$ có $S(10) = 10$ là lớn nhất.
100	10	
10	22	
864	118	
130056192	90	
1000000000	2	
2	333333009	
999999018		

Bài 12. Ước số của Hai Số Nguyên

xydivisors.cpp

Gần đây, bạn nhận được hai số nguyên dương x và y . Bạn đã quên chúng, nhưng bạn nhớ một danh sách các số đã được xáo trộn, chứa tất cả các ước số của x (bao gồm cả 1 và x) và tất cả các ước số của y (bao gồm cả 1 và y). Nếu d là ước số chung của cả x và y , thì d xuất hiện hai lần trong danh sách.

Ví dụ: nếu $x=4$ và $y=6$ thì danh sách cho trước có thể là bất kỳ hoán vị nào của danh sách $[1, 2, 4, 1, 2, 3, 6]$. Một số danh sách có thể là: $[1, 1, 2, 4, 6, 3, 2]$, $[4, 6, 1, 1, 2, 3, 2]$ hoặc $[1, 6, 3, 2, 4, 1, 2]$.

Yêu cầu: Khôi phục lại hai số nguyên dương thích hợp x và y để tạo ra cùng một danh sách các ước số (có thể ở thứ tự khác).

Đảm bảo rằng có câu trả lời, tức là danh sách các ước số cho trước tương ứng với một số cặp số nguyên dương x và y .

Dữ liệu vào:

Dòng đầu tiên chứa một số nguyên n ($2 \leq n \leq 128$) - số lượng ước số của x và y . Dòng thứ hai chứa n số nguyên d_1, d_2, \dots, d_n ($1 \leq d_i \leq 10^4$), trong đó d_i là ước số của x hoặc y . Nếu một số là ước số chung của cả x và y thì số đó xuất hiện hai lần trong danh sách.

Dữ liệu ra:

In ra hai số nguyên dương x và y - các số mà danh sách các ước số hợp nhất của chúng là một hoán vị của danh sách các số nguyên cho trước. Đảm bảo rằng có câu trả lời. Giả sử $x > y$.

Ví dụ:

Input	Output	
10	20 8	
10 2 8 1 2 4 1 20 4 5		

Bài 13. Số nguyên tố cùng nhau

Coprime.cpp

Cho một mảng gồm n số nguyên dương a_1, a_2, \dots, a_n ($1 \leq a_i \leq 1000$). Tìm giá trị lớn nhất của $i + j$ sao cho a_i và a_j là hai số nguyên tố cùng nhau, hoặc trả về -1 nếu không tìm thấy cặp số nào thỏa mãn.

Ví dụ: Xét mảng [1,3,5,2,4,7,7]. Giá trị lớn nhất của $i + j$ có thể đạt được là 5+7, vì $a_5=4$ và $a_7=7$ là hai số nguyên tố cùng nhau.

Lưu ý: Hai số nguyên p và q được gọi là **nguyên tố cùng nhau** nếu ước số chung lớn nhất của chúng là 1.

Dữ liệu vào:

- Dòng đầu tiên chứa một số nguyên n ($2 \leq n \leq 2 \cdot 10^5$) – độ dài của mảng.
- Dòng tiếp theo chứa n số nguyên dương a_1, a_2, \dots, a_n ($1 \leq a_i \leq 1000$) – các phần tử của mảng.

Dữ liệu ra: In ra một số nguyên duy nhất – giá trị lớn nhất của $i+j$ thỏa mãn điều kiện, hoặc -1 nếu không có cặp số nào thỏa mãn.

Ví dụ:

Input	Output	Giải thích
3 3 2 1	6	Chọn $i=j=3$
7 1 3 5 2 4 8 8	10	Chọn $i=3, j=7$
5 1 2 3 4 5	9	Chọn $i=4, j=5$
3 2 2 4	-1	Không tồn tại

Subtask 1: 50% test có $n \leq 10^3$

Subtask 2: 50% test còn lại có $n \leq 2 \cdot 10^5$.

Bài 14. Tích của ba số

Threenumbers.cpp

Cho một số nguyên dương n . Nhiệm vụ của bạn là tìm 3 số nguyên dương khác nhau a, b, c sao cho khi nhân 3 số đó với nhau thì bằng n . Nếu không tìm được bộ ba số nào thỏa mãn, hãy trả lời "NO", nếu tồn tại ghi "YES".

Ví dụ:

- Nếu $n = 24$, một đáp án có thể là $a = 2, b = 3, c = 4$ vì $2 * 3 * 4 = 24$.
- Nếu $n = 7$, không có cách nào chia 7 thành tích của 3 số nguyên dương khác nhau lớn hơn 1 nên đáp án là "NO".

Điều kiện:

- a, b, c phải khác nhau.
- a, b, c phải lớn hơn hoặc bằng 2.

Dữ liệu vào:

- Dòng đầu tiên chứa số lượng trường hợp kiểm tra t ($t \leq 100$).
- Mỗi trường hợp kiểm tra chứa một số nguyên n ($2 \leq n \leq 10^9$).

Dữ liệu ra:

- In ra "YES" và 3 số a, b, c nếu tìm được.
- In ra "NO" nếu không tìm được.

Ví dụ:

Input	Output	Giải thích
5	YES	Ta có: $64=2.4.8$
64	NO	$12345=3.5.823$

32	NO	
97	NO	
2	YES	
12345		

Bài 15. Dãy số đẹp

Beauty.cpp

Cho một dãy số nguyên a_1, a_2, \dots, a_n và một số nguyên k . Bạn cần thực hiện ít nhất các phép toán để biến dãy số này thành một dãy đẹp.

Trước khi thực hiện các phép toán, bạn có thể hoán vị các phần tử trong dãy theo ý muốn. Với mỗi phép toán, bạn có thể làm như sau:

- Chọn một chỉ số $1 \leq i \leq n$.
- Gán $a_i = a_i + k$.

Một dãy số b_1, b_2, \dots, b_n được gọi là đẹp nếu $b_i = b_{n-i+1}$ với mọi $1 \leq i \leq n$.

Hãy tìm số lượng phép toán tối thiểu cần thiết để biến dãy số thành một dãy đẹp, hoặc báo rằng điều đó là không thể.

Dữ liệu vào:

Mỗi test bao gồm nhiều bộ dữ liệu vào. Dòng đầu tiên chứa một số nguyên duy nhất t ($1 \leq t \leq 10^4$) - số lượng bộ dữ liệu vào. Tiếp theo là mô tả của các bộ dữ liệu.

Dòng đầu tiên của mỗi bộ dữ liệu vào chứa hai số nguyên n và k ($1 \leq n \leq 10^5$, $1 \leq k \leq 10^9$) - kích thước của dãy a và số k trong đề bài.

Dòng thứ hai của mỗi bộ dữ liệu vào chứa n số nguyên a_1, a_2, \dots, a_n ($1 \leq a_i \leq 10^9$) - các phần tử của dãy a .

Đảm bảo rằng tổng của n trên tất cả các bộ dữ liệu vào không vượt quá $2 \cdot 10^5$.

Dữ liệu ra:

Với mỗi bộ dữ liệu vào, in ra số lượng phép toán tối thiểu cần thiết để biến dãy số thành một dãy đẹp, hoặc -1 nếu điều đó là không thể.

Ví dụ:

Input	Output
11	0
1 1000000000	83966524
1	1
2 1	4
624323799 708290323	6
3 1	1
3 2 1	48
4 1	-1
7 1 5 3	
5 1	
11 2 15 7 10	
7 1	
1 8 2 16 8 16 31	
13 1	

2 1 1 3 3 11 12 22 45 777 777 1500 74 10 2 1 2 1 2 1 2 1 2 2	
--	--

Giải thích:

- Bộ dữ liệu đầu tiên, mảng đã đẹp.
- Bộ dữ liệu thứ hai, bạn có thể hoán vị mảng trước các phép toán và thực hiện phép toán với chỉ số $i = 1$ cho 83966524 lần.
- Bộ dữ liệu thứ ba, bạn có thể hoán vị mảng a và làm cho nó bằng với $[2, 3, 1]$. Sau đó áp dụng phép toán với chỉ số $i = 3$ để nhận được mảng $[2, 3, 2]$, mảng này là đẹp.
- Bộ dữ liệu thứ tám, không có tập hợp các phép toán và không có cách nào để hoán vị các phần tử để làm cho mảng đẹp.
- Bộ dữ liệu thứ chín, mảng đã đẹp.

Subtask 1: Có 70% test có $t * n < 5000$

Subtask 2: Có 30% test có $t * n \leq 200000$.

Bài 16. Chia sau khi cộng thêm K

SPLIT.cpp

Bạn có một bảng đen với n số nguyên dương a_1, a_2, \dots, a_n . Bạn cũng được cho một số nguyên dương k . Bạn có thể thực hiện một số (có thể là 0) lần thao tác sau:

- Chọn một số x trong dãy.
- Xóa x khỏi dãy và thay bằng hai số nguyên dương y, z sao cho $y + z = x + k$.

Liệu có thể làm cho tất cả các số trên bảng đen bằng nhau không? Nếu có, thì số lượng tối thiểu các thao tác bạn cần là bao nhiêu? Nếu không thì in ra -1.

Dữ liệu vào:

Mỗi test case bao gồm nhiều bộ dữ liệu. Dòng đầu tiên chứa số lượng test case t ($1 \leq t \leq 10^4$). Tiếp theo là mô tả của các test case.

Dòng đầu tiên của mỗi test case chứa hai số nguyên n, k ($1 \leq n \leq 2 * 10^5, 1 \leq k \leq 10^{12}$) - số lượng số nguyên ban đầu trên bảng đen và hằng số k .

Dòng thứ hai của mỗi test case chứa n số nguyên a_1, a_2, \dots, a_n ($1 \leq a_i \leq 10^{12}$) - trạng thái ban đầu của bảng đen.

Đảm bảo rằng tổng của n trên tất cả các test case không vượt quá $2 * 10^5$.

Dữ liệu ra:

Với mỗi test case, in ra một dòng duy nhất chứa một số nguyên: số lượng tối thiểu các thao tác bạn cần để làm cho tất cả các số trên bảng đen bằng nhau, hoặc -1 nếu điều đó là không thể.

Ví dụ:

Input	Output	Giải thích
6	3	<ul style="list-style-type: none"> Trong test case đầu tiên, $k = 1$. Bạn có thể làm cho tất cả các số trên bảng đen bằng 2 bằng các thao tác sau: Xóa $x = 4$ và viết $(y, z) = (2, 3)$. Ta được $\{3, 2, 3\}$. Xóa $x = 3$ và viết $(y, z) = (2, 2)$. Ta được
2 1	1	
3 4	4	
2 3	-1	

7 11	-1	{2, 2, 2, 3}. Xóa $x = 3$ và viết $(y, z) = (2, 2)$. Ta được {2, 2, 2, 2, 2}. Biến tất cả các số bằng nhau sau 3 thao tác. Có thể chứng minh không thể làm cho tất cả các số bằng nhau trong ít hơn 3 thao tác. • Trong test case thứ hai, $k = 3$. Bạn có thể làm cho tất cả các số bằng 7 như sau: Xóa $x=11$ và viết $(y, z) = (7, 7)$. Ta được: {7, 7, 7}.
3 10	0	
100 40 100		
2 1		
1 2		
2 2		
1 2		
1 327869541		
327869541		

Subtask 1: 70% test có $t * n \leq 500$

Subtask 2: 30% test còn lại.

Bài 17. Hàng rào vô cực

Fence.cpp

Bạn là một lãnh đạo nổi dậy đang lên kế hoạch khởi nghĩa ở đất nước của mình. Tuy nhiên, chính phủ độc ác đã phát hiện ra kế hoạch của bạn và đưa ra hình phạt dưới hình thức lao động cải tạo.

Bạn phải sơn một hàng rào gồm 10^{100} tấm ván bằng hai màu theo cách sau (giả sử các tấm ván được đánh số từ trái sang phải từ 0):

- Nếu chỉ số của tấm ván chia hết cho r (các tấm ván như vậy có chỉ số 0, r , $2r$, v.v.), bạn phải sơn màu đỏ.
- Nếu chỉ số của tấm ván chia hết cho b (các tấm ván như vậy có chỉ số 0, b , $2b$, v.v.), bạn phải sơn màu xanh.
- Nếu chỉ số chia hết cho cả r và b , bạn có thể chọn màu để sơn tấm ván.
- Nếu không, bạn không cần sơn tấm ván (và việc sử dụng sơn trên đó là cấm).

Hơn nữa, chính phủ đã thêm một hạn chế bổ sung để làm cho hình phạt của bạn tệ hơn. Hãy liệt kê tất cả các tấm ván đã sơn của hàng rào theo thứ tự tăng dần: nếu có k tấm ván liên tiếp cùng màu trong danh sách này, thì chính phủ sẽ tuyên bố rằng bạn đã thất bại trong công việc và xử tử bạn ngay lập tức. Nếu bạn không sơn hàng rào theo bốn điều kiện đã nêu trên, bạn cũng sẽ bị xử tử.

Câu hỏi là: liệu bạn có thể hoàn thành công việc (thời gian không quan trọng) hay việc xử tử là không thể tránh khỏi và bạn cần trốn thoát bằng mọi giá.

Dữ liệu vào:

Dòng đầu tiên chứa một số nguyên duy nhất T ($1 \leq T \leq 1000$) - số lượng test case.

T dòng tiếp theo chứa mô tả của các test case - mỗi dòng một test case. Mỗi test case chứa ba số nguyên r, b, k ($1 \leq r, b \leq 10^9, 2 \leq k \leq 10^9$) - các hệ số tương ứng.

Dữ liệu ra:

In ra T từ - mỗi dòng một từ. Với mỗi test case, in ra 0 (nếu việc xử tử là không thể tránh khỏi hoặc 1 nếu ngược lại.

Ví dụ:

Input	Output	
4	1	
1 1 2	0	

2 10 4	1	
5 2 3	1	
3 2 2		

<https://codeforces.com/problemset/problem/1612/D>