**HƯỚNG DẪN GIẢI BÀI CONTEST 3**

**Thuật ngữ:**

· gcd : ước chung lớn nhất

· Lcm : Bội chung nhỏ nhất

**Một số kiến thức hỗ trợ:**

· Số lượng số chia hết cho x trong đoạn [1..n] là n div x

· Số lượng số chia hết cho x trong đoạn [L..R] là (R div x) – ((L-1) div x)

· Số lượng số chia hết cho a hoặc b trong đoạn [1..n]:

Là (n div a) + (n div b) – (n div lmc(a,b))

**BÀI A: CUTSQARE - CẮT HÌNH VUÔNG**

- Gọi cạnh hình vuông cắt được là x. Khi đó x phải là ước của A và là ước của B mà muốn x lớn nhất thì x phải là gcd của A và B.

- Nên đáp án là: A/x \* B/x

**BÀI B: GCD2ONE - XOÁ SỐ ĐỂ GCD LÀ 1**

- Gọi u = gcd(A1, A2, …, An)

- Nếu u = 1 thì tất nhiên là không cần xoá à Đáp án là 0

- Nếu u > 1 thì không có cách xoá, vì cho dù xoá bất cứ số nào đi nữa thì gcd không thể bằng 1. Nên đáp án là -1.

**BÀI C: MAXSUMDIV - ĐẾM BỘI**

- Gọi L là lcm(a, b).

- Gọi d1 là số lượng bội của a trong đoạn [1..n]. Khi đó d1 = n div a

- Gọi d2 là số lượng bội của b trong đoạn [1..n]. Khi đó d2 = n div b

- Gọi d3 là số lượng bội của a và b trong đoạn [1..n]. Khi đó d3 = n div lcm

- Đáp án sẽ là: d1\*x + d2\*y – d3\*min(x,y) ([Theo nguyên lý bao hàm loại trừ](https://vnoi.info/wiki/translate/he/Number-Theory))

- Trừ đi d3 vì d3 đã được tính hai lần trong d1 và d2. Nhưng ta muốn lấy giá trị max nên ta loại đi phần giá trị đã được tính nhỏ hơn tức là d3\*min(x,y).

**BÀI D: CHINUOC - ĐẾM SỐ CÓ 9 ƯỚC**

- Đầu tiên ta duyệt vét in ra các số có 9 ước để kiểm tra và rút ra những nhận xét để có thể giải quyết bài.

- Một số có chín ước nếu nó thoả mãn:

· Bình phương của tích hai số nguyên tố: Ví dụ 36 = (2\*3)^2

· Hoặc là x^8 với x là số nguyên tố. Ví dụ 2^8

- Trường hợp 1: Ta vét hai for trên danh sách các số nguyên tố ≤ sqrt(n)

- Trường hợp 2: Ta kiểm tra 7 số nguyên tố đầu tiên.

**Bài E: QUADRUGCD - BỘ TỨ CÓ GCD LÀ K**

- Cách vét đơn giản nhất là 4 for để kiểm tra. Với n ≤ 5000 ta có thể nghĩ đến tối ưu chỉ còn lại 2 for.

- Một cách tối ưu đơn giản là ta duyệt đánh dấu trước 2 for. Đoạn lệnh sau sẽ đếm số cặp (i,j) có gcd(i,j).

foru(i,L,R) foru(j,L,R) dem[G[i][j]]++;

- Sau đó với cặp (**x, y**) x, y ≤ R thoả mãn: gcd(x,y)=k thì đáp án số bộ tứ tương ứng với cặp này sẽ là dem[x]\*dem[y].

- Bài này time chặt nên cần tính trước bảng ước chung lớn nhất G[i][j] = gcd(i,j) theo công thức:

· G[i][0] = G[i][i] = i

· G[i][j] = G[j][i mod j]

- Cảm ơn bạn vinowadays đã hướng dẫn công thức để tính trước bảng G[i][j].

- Bài này có một cách làm khác sử dụng kiến thức toán và nguyên lý bù trừ.

**BÀI F: CUTCAKE - CẮT BÁNH SINH NHẬT**

- Gọi g = gcd(a[i] – a[i-1], 360-a[n]) với i = 1..n

- Đáp án sẽ là (360 div g) - n - 1

**BÀI G: LARGNDIGI - TÌM BỘI SỐ**

- Gọi L = lcm(a,b,c)

- Gọi x = 10^n

- Gọi u = x mod L

- Nếu x – u >= 10^(n-1) thì đáp án chính là x – u. Còn không thì ko có đáp án.

**BÀI H: DIVARANGE - TÌM BỘI TRONG KHOẢNG**

- Gọi K = lcm(a[1], …, a[n])

- Đáp án là : (R div K) – ((L-1) div K)

**Bài I: NTHDIVISI - SỐ CHIA HẾT THỨ N**

- Đáp án số X chia hết thứ N nếu có sẽ là trong đoạn [lcm, 10^18]

- Ta thấy N càng lớn thì X càng lớn vậy thì ta có thể chặt nhị phân trên đoạn này để tìm X.

- Khi chặt nhị phân: Với một số nguyên mid ta kiểm tra xem từ [1..mid] có đủ N số là bội của A và B hay không.

+ Số bội là: dem = (mid div A) + (mid div b) – (mid div lcm)

+ Nếu dem > n thì ta giảm khoảng tìm kiếm còn không thì tăng khoảng tìm kiếm.

**BÀI J: GCDALTN - GCD LUỸ THỪA**

- GCD(a^N, b) = GCD(b, a^n mod b)

- Vậy nên ta có thể tính trước U = a^n mod b ([Luỹ thừa nhanh](https://vietcodes.github.io/algo/squaring-exponentiation))

- Đáp án là gcd(u, b)

**Bài K: JUMP2GCD - BƯỚC NHẢY ALPHA**

- Từ vị trí K có thể jumps đến vị trí x khi: x = k + i\*d1 + j\*d2 (i, j nguyên)

Gọi gcd = \_\_gcd(d1, d2) khi đó ta có d1 = m1\*gcd, d2 = m2\*gcd

x = k + i\*m1\*gcd + j\*m2\*gcd

= k + gcd\*(i\*m1 + j\*m2)

<=> x - k = gcd\*(i\*m1 + j\*m2)

<=> (i\*m1 + j\*m2) = (x - k)/gcd

Nếu (x - k) chia hết cho gcd: Gọi số nguyên U = (x - k) / gcd

<=> (i\*m1 + j\*m2) = u

Đây là phương trình có dạng ax + by = c Luôn luôn có vô số nghiệm

==> có thể đến được x