



BÀI TẬP LẬP TRÌNH NÂNG CAO



1☀. Tổng Ước chung lớn nhất

Cho số nguyên dương K . Hãy tính $X = \sum_{a=1}^K \sum_{b=1}^K \sum_{c=1}^K \gcd(a, b, c)$.

Dữ liệu cho trong file **SUMGCD.INP** gồm số nguyên dương K ($K \leq 200$).

Kết quả ghi ra file **SUMGCD.OUT** là số X tính được.

Ví dụ:

SUMGCD.INP	SUMGCD.OUT
2	9

Giải thích:

$$\begin{aligned} &\gcd(1,1,1) + \gcd(1,1,2) + \gcd(1,2,1) + \gcd(1,2,2) + \gcd(1,1,1) + \gcd(1,1,2) + \gcd(1,2,1) + \\ &\gcd(1,2,2) + \gcd(2,1,1) + \gcd(2,1,2) + \gcd(2,2,1) + \gcd(2,2,2) + \gcd(2,1,1) + \gcd(2,1,2) + \\ &\gcd(2,2,1) + \gcd(2,2,2) = 1 + 1 + 1 + 1 + 1 + 1 + 1 + 1 + 2 = 9. \end{aligned}$$



2☀. Chuyển tự động

Hệ thống sản xuất tự động gồm N dây chuyền sản xuất. Các dây chuyền được đánh số từ 1, 2, 3, ..., N . Điều thú vị là dây chuyền sản xuất không thực hiện tuần tự dây chuyền 1 rồi đến dây chuyền 2, ..., mà dây chuyền thứ i chuyển đến dây chuyền A_i ($i \neq A_i$).

Yêu cầu: Tính xem, bắt đầu từ dây chuyền 1, sau K lần chuyển dây chuyền thì hệ thống đang thực hiện dây chuyền nào?

Dữ liệu cho trong file **AutoK.Inp** gồm:

- Dòng đầu ghi 2 số nguyên dương N và K .
- Dòng thứ 2 ghi N số nguyên dương A_1, A_2, \dots, A_N .

Kết quả ghi ra file **AutoK.Out** số u là số hiệu của dây chuyền mà hệ thống đang thực hiện sau K lần dịch chuyển dây chuyền.

Ví dụ:

AutoK.Inp	AutoK.Out	Giải thích
4 5 3 2 4 1	4	$1 \rightarrow 3 \rightarrow 4 \rightarrow 1 \rightarrow 3 \rightarrow 4$

Giới hạn:

- $2 \leq N \leq 2 \times 10^5$;
- $1 \leq A_i \leq N$;
- $1 \leq K \leq 10^{18}$.

**3☀. Số lập phương**

Ta gọi e là một số lập phương nếu $e = u^3$. Ví dụ, $8 = 2^3$, $27 = 3^3$. Cho hai số nguyên x và y ($x \leq y$). Tính số các số lập phương thuộc đoạn $[x; y]$.

Dữ liệu cho trong file **NumCube.Inp** gồm 2 số nguyên x và y ($x \leq y$).

Kết quả ghi ra file **NumCube.Out** là số các số lập phương thuộc đoạn $[x; y]$.

Ví dụ:

NumCube.Inp	NumCube.Out	Giải thích
0 30	4	Có 4 số lập phương: 0, 1, 8, 27.

Giới hạn:

$$-10^{18} \leq x \leq y \leq 10^{18};$$

**4☀. Tách số**

Cho số N có k chữ số, $N = \overline{e_1 e_2 \dots e_k}$. Ta gọi $s_i = \overline{e_1 e_2 \dots e_i} + \overline{e_{i+1} \dots e_k}$ với $i = 1, 2, 3, \dots, k-1$.

Yêu cầu: Tính tổng $s_1 + s_2 + \dots + s_{k-1}$.

Dữ liệu ghi trong file **TachSo.Inp** gồm một số nguyên dương N ($10 \leq N \leq 10^{18}$).

Kết quả ghi ra file **TachSo.Out** là tổng $s_1 + s_2 + \dots + s_{k-1}$.

Ví dụ:

TachSo.Inp	TachSo.Out	Giải thích
123	39	$s_1 = 1 + 23$; $s_2 = 12 + 3$

**5☀. Dãy số 7..7**

Nhiều người không thích số 7, nhưng Bé Vy thì lại rất thích số 7. Một hôm, cô bé viết lên tờ giấy một dãy số: 7, 77, 777, 7777, ... (các số chỉ gồm các chữ số 7). Điều mà cô bé quan tâm lúc này là cần tìm số nhỏ nhất trong dãy chia hết cho K .

Do số có thể rất lớn nên Vy nhờ các anh chị lập trình tìm số hử mình.

Dữ liệu cho trong dãy **NUM7.INP** gồm số nguyên dương K ($K \leq 10^6$).

Kết quả ghi ra file **NUM7.OUT** là số nhỏ nhất trong dãy chia hết cho K . Nếu không có số nào chia hết cho K thì ghi -1.

Ví dụ:

NUM7.INP	NUM7.OUT
101	7777
2	-1