

Problems in Elementary Computer Science – Bài Tập Tin Học Sơ Cấp

Nguyễn Quân Bá Hồng*

Ngày 25 tháng 1 năm 2024

Bài toán 1 ([Huy24], p. 6, số thân thiện – friendly number). (a) Tìm tất cả các số tự nhiên có 2 chữ số mà khi đảo trật tự của 2 chữ số đó sẽ thu được 1 số nguyên tố cùng nhau với số đã cho. (b) Mở rộng 2 thành $n \in \mathbb{N}, n \geq 2$.

Bài toán 2 ([Huy24], p. 13, cấp số cộng – arithmetic progression). Tìm các số tự nhiên lẻ gồm 3 chữ số. 3 chữ số này, theo thứ tự từ trái qua phải tạo thành 1 cấp số cộng.

Bài toán 3 ([Huy24], p. 17, cấp số nhân – geometric progression). Tìm các số tự nhiên gồm 3 chữ số. 3 chữ số này, theo thứ tự từ trái qua phải tạo thành 1 cấp số nhân với công bội $q \in \mathbb{N}^*$.

Bài toán 4 ([Tru23], 1., p. 13, HSG Lớp 10 Vĩnh Phúc 2020–2021, Square – Hình vuông). (a) Cho n điểm có tọa độ là các số nguyên trên hệ trục tọa độ Oxy. Tìm diện tích hình vuông nhỏ nhất có các cạnh song song với các trục tọa độ sao cho tất cả các điểm đã cho đều thuộc hình vuông đó (điểm nằm trên cạnh hình vuông cũng được coi là thuộc hình vuông đó).

- **Input.** Dòng 1: chứa số nguyên dương $n \in \mathbb{N}^*, 2 \leq n \leq 20$, là số lượng điểm có tọa độ là các số nguyên. n dòng tiếp theo, mỗi dòng ghi 2 số nguyên $x, y \in \mathbb{Z}, 1 \leq x, y \leq 100$, là tọa độ của mỗi điểm.
- **Output.** Ghi diện tích hình vuông nhỏ nhất tìm được.
- **Sample.**

square.inp	square.out
3	16
3 4	
5 7	
4 3	

(b) Mở rộng bài toán từ ‘hình vuông’ sang ‘hình chữ nhật’, & từ ‘tọa độ nguyên’ sang ‘tọa độ thực’: (i) Cho n điểm có tọa độ là các số thực trên hệ trục tọa độ Oxy. Tìm diện tích hình vuông, hình chữ nhật “nguyên”, hình chữ nhật “nguyên” nhỏ nhất có các cạnh song song với các trục tọa độ sao cho tất cả các điểm đã cho đều thuộc hình chữ nhật đó (điểm nằm trên cạnh hình chữ nhật cũng được coi là thuộc hình chữ nhật đó), trong đó hình vuông, hình chữ nhật “nguyên” lần lượt là các hình vuông & hình chữ nhật có các tọa độ của 4 đỉnh là 8 số nguyên.

Bài toán 5 ([Tru23], 2., pp. 13–14, HSG Lớp 10 Vĩnh Phúc 2020–2021, Divisible by 3 – Chia hết cho 3). Cho dãy a gồm n số nguyên dương. Cho biết có bao nhiêu cặp số trong dãy có tổng chia hết cho 3, i.e., đếm xem có bao nhiêu cặp chỉ số $i, j, 1 \leq i < j \leq n$, sao cho $a_i + a_j \div 3$.

- **Input.** Dòng 1: 1 số nguyên duy nhất $n, 1 \leq n \leq 10^5$. Dòng 2: Ghi n số nguyên dương $a_1, a_2, \dots, a_n, 1 \leq a_i \leq 10^5, \forall i = 1, 2, \dots, n$, là các phần tử của dãy.
- **Output.** 1 dòng duy nhất ghi số lượng cặp số của dãy a có tổng chia hết cho 3.
- **Sample.**

div3.inp	div3.out	Giải thích
5 3 6 9 12	3	3 cặp số tìm được có chỉ số: (1, 4), (2, 3), (3, 5).
4 3 6 9 12	6	6 cặp số tìm được có chỉ số: (1, 2), (1, 3), (1, 4), (2, 3), (2, 4), (3, 4).

Bài toán 6 ([Tru23], 3., p. 14, HSG Lớp 10 Vĩnh Phúc 2020–2021, Delete element – Xóa phần tử). Cho dãy gồm n số nguyên a_1, a_2, \dots, a_n với $1 \leq a_i \leq 3, \forall i = 1, 2, \dots, n$. Có bao nhiêu cách để xóa đi 1 số phần tử của dãy (không xóa phần tử nào cũng được coi là 1 cách) mà vẫn giữ nguyên thứ tự ban đầu để được 1 dãy mới thỏa mãn 2 yêu cầu sau: (i) Dãy còn ít nhất 3 phần tử. (ii) Phần tử đầu tiên của dãy có giá trị 1, tiếp theo là 1 số phần tử có giá trị là 2 (ít nhất có 1 số 2), & kết thúc bằng đúng 1 phần tử có giá trị là 3. E.g., các dãy 1, 2, 2, 3 & 1, 2, 3 thỏa mãn yêu cầu, các dãy 1, 2, 3, 3 & 1, 1, 2, 3 không thỏa mãn yêu cầu.

*e-mail: nguyenquanbahong@gmail.com, website: <https://nqbh.github.io>, Ben Tre City, Vietnam.

- **Input.** Dòng 1: 1 số nguyên dương $n \in \mathbb{N}^*$, $n \leq 10^6$, là số lượng phần tử của dãy. Dòng 2: Ghi n số nguyên dương a_1, a_2, \dots, a_n là giá trị của các phần tử của dãy ban đầu.
- **Output.** Gồm 1 dòng duy nhất là số cách xóa để được dãy mới thỏa mãn yêu cầu của đề bài. Do số lượng cách xóa phần tử có thể rất lớn nên chỉ cần ghi ra số lượng cách xóa sau khi chia lấy dư cho $10^9 + 7$.
- **Sample.**

delete_element.inp	delete_element.out
8 1 2 1 2 3 1 2 3	15

Bài toán 7 ([Tru23], 1., p. 15, HSG Lớp 11 Vĩnh Phúc 2020–2021, Game button – Trò chơi bấm nút). Người chơi đang tham gia 1 trò chơi như sau: Có 2 nút bấm A, B, trên nút A có ghi số m_A , trên nút B có ghi số m_B . Ở mỗi lượt chơi, người chơi phải chọn bấm 1 trong 2 nút & sẽ nhận được số điểm thưởng bằng với số ghi trên nút đó, sau đó số trên nút bấm giảm đi 1 đơn vị. Hỏi sau 2 lượt chơi, số điểm thưởng lớn nhất mà người chơi có thể nhận được là bao nhiêu?

- **Input.** 1 dòng duy nhất ghi 2 số nguyên dương m_A, m_B với $3 \leq A, B \leq 20$, tương ứng với 2 số ghi trên 2 nút A & B.
- **Output.** Ghi số điểm thưởng lớn nhất mà người chơi có thể nhận được sau 2 lượt chơi.
- **Sample.**

game_button.inp	game_button.out	Giải thích
5 3	9	Bấm 2 lần nút A & sẽ có tổng điểm thưởng: $5 + 4 = 9$.

Bài toán 8 ([Tru23], 2., p. 15, HSG Lớp 11 Vĩnh Phúc 2020–2021, Count number – Đếm số). Cho 4 số nguyên dương a, b, c, d . Đếm xem có bao nhiêu số nguyên dương $x \in \mathbb{N}^*$ thỏa mãn các điều kiện sau: (i) $a \leq x \leq b$. (ii) $x \nmid c$. (iii) $x \nmid d$.

- **Input.** 1 dòng duy nhất ghi 4 số a, b, c, d , $1 \leq a, b \leq 10^{18}$, $1 \leq c, d \leq 10^9$.
- **Output.** 1 dòng duy nhất ghi số lượng số nguyên dương $x \in \mathbb{N}^*$ thỏa mãn điều kiện đề bài.
- **Sample.**

count_number.inp	count_number.out	Giải thích
4 9 2 3	2	Chỉ có số 5 & 7 thỏa mãn điều kiện đề bài.

Bài toán 9 ([Tru23], 3., p. 16, HSG Lớp 11 Vĩnh Phúc 2020–2021, Reverse & reverse – Lật qua lật lại). Cho dãy a gồm $n \in \mathbb{N}^*$ phần tử $1, 2, \dots, n$. Người ta thực hiện trên dãy số này đúng k lần 2 thao tác sau: (i) Đầu tiên, đảo ngược thứ tự (lật đối xứng) đoạn phần tử có chỉ số từ u đến v . (ii) Tiếp theo, đảo ngược thứ tự (lật đối xứng) đoạn phần tử có chỉ số từ l đến r . Với u, v, l, r là các hằng số cho trước. Đưa ra dãy a sau khi thực hiện k lần 2 thao tác nói trên.

- **Input.** Dòng 1: 2 số nguyên dương $n, k \in \mathbb{N}^*$, $1 \leq n \leq 100$, $1 \leq k \leq 10^9$. Dòng 2: gồm 2 số nguyên dương u, v , $1 \leq u < v \leq n$. Dòng 3: gồm 2 số nguyên dương l, r , $1 \leq l < r \leq n$.
- **Output.** Ghi trên n dòng, dòng thứ i ghi giá trị của phần tử thứ i của dãy a sau khi thực hiện k lần 2 thao tác nói trên, $\forall i = 1, 2, \dots, n$.
- **Sample.**

reverse_reverse.inp	reverse_reverse.out	Giải thích
7 2 2 5 3 7	1 2 4 3 5 7 6	Dãy ban đầu: 1 2 3 4 5 6 7 Lần 1: 1 5 4 3 2 6 7 1 5 7 6 2 3 4 Lần 2: 1 2 6 7 5 3 4 1 2 4 3 5 7 6

Bài toán 10 ([Tru23], 3., p. 17, HSG Lớp 12 Vĩnh Phúc 2020–2021, Max gift – Chọn quà mắc nhất). Cuối năm công ty tổ chức phát quà cho nhân viên. Có $n \in \mathbb{N}^*$ gói quà với giá trị khác nhau được xếp liên tiếp thành 1 hàng, trong đó gói quà thứ i có giá trị là a_i . Mỗi nhân viên chỉ được chọn 2 gói quà liên tiếp. Mr. Bean là người may mắn được chọn đầu tiên. Giúp Mr. Bean chọn ra 2 gói quà liên tiếp có giá trị lớn nhất.

- Input. Dòng 1: chứa số nguyên dương $n \in \mathbb{N}^*$, $2 \leq n \leq 10^6$. Dòng 2: Giá trị của n gói quà, $1 \leq a_i \leq 10^3$, $\forall i = 1, 2, \dots, n$, mỗi giá trị cách nhau bởi dấu cách.
- Output. 1 dòng duy nhất chứa tổng giá trị quà lớn nhất chọn được.
- Sample.

max_gift.inp	max_gift.out
5 1 3 5 4 2	9

Bài toán 11 ([Tru23], 2., pp. 17–18, HSG Lớp 12 Vĩnh Phúc 2020–2021, Decrease value – Giảm giá trị). 1 ngày rảnh rỗi, Mr. Bean chơi trò chơi với các con số. Mr. Bean lấy 1 số nguyên dương $n \in \mathbb{N}^*$ rồi thực hiện không giới hạn số lần thao tác “Chọn 1 chữ số x của n rồi giảm n đi x đơn vị”. Hỏi Mr. Bean phải thực hiện ít nhất bao nhiêu thao tác như vậy để giảm số n về 0. E.g., $n = 27$, Mr. Bean sẽ thực hiện 5 thao tác để biến đổi: (i) Chọn $x = 7 \rightarrow n = 27 - 7 = 20$. (ii) Chọn $x = 2 \rightarrow n = 20 - 2 = 18$. (iii) Chọn $x = 8 \rightarrow n = 18 - 8 = 10$. (iv) Chọn $x = 1 \rightarrow n = 10 - 1 = 9$. (v) Chọn $x = 9 \rightarrow n = 9 - 9 = 0$.

- Input. 1 dòng: 1 số nguyên dương duy nhất n , $1 \leq n \leq 10^6$.
- Output. 1 dòng duy nhất ghi số thao tác ít nhất để biến đổi n về 0.
- Sample.

decrease_value.inp	decrease_value.out
27	5

Bài toán 12 ([Tru23], 3., p. 18, HSG Lớp 12 Vĩnh Phúc 2020–2021, Difference degree of substrings – Xâu con phân biệt). 1 lần Mr. Bean được bạn gái gửi cho 1 dãy ký tự S độ dài n chỉ gồm các chữ cái in hoa ('A', ..., 'Z'). Bạn gái nhờ Mr. Bean xác định “độ phân biệt” của dãy ký tự trên. Trong đó độ phân biệt của dãy ký tự là số nguyên dương l nhỏ nhất sao cho tất cả các xâu con của S độ dài l là đôi một phân biệt. E.g., với $n = 7$, $S = \text{'ABCDABC'}$ thì $l = 4$ do tất cả các xâu con độ dài 4 đều phân biệt. Giúp Mr. Bean việc đó.

- Input. Dòng 1: số nguyên dương $n \in \mathbb{N}^*$, $n \leq 100$. Dòng 2: chứa xâu ký tự S .
- Output. Gồm 1 dòng duy nhất ghi 1 số nguyên duy nhất là “độ phân biệt” của dãy ký tự S .
- Sample.

diff_substring.inp	diff_substring.out
7 ABCDABC	4

Bài toán 13 ([Tru23], 4., p. 18, HSG Lớp 12 Vĩnh Phúc 2020–2021, Ants meet – Kiến tha mồi). Trên đường đi làm về Mr. Bean quan sát thấy 2 ổ kiến cách nhau 1 khoảng l đơn vị. Các con kiến đang tha mồi về 2 tổ trên đường thẳng nối 2 tổ kiến với nhau. Các con kiến khi tha mồi về tổ nào thì ở lại tổ đó. Nếu 2 con kiến gặp nhau trên đường đi thì cả 2 sẽ đổi hướng di chuyển.

Giả sử đường nối giữa 2 tổ kiến được gắn tọa độ từ 0 đến l . Tổ thứ nhất ở vị trí 0 & tổ thứ 2 ở vị trí l . Ở thời điểm Mr. Bean quan sát có n con kiến đang tha mồi về tổ. Con thứ i xuất phát ở tọa độ x_i , mang lượng mồi khối lượng w_i & có hướng di chuyển d_i . Nếu $d_i = 1$ thì con kiến thứ i đang di chuyển theo hướng 0 về l , $d_i = -1$ thì con kiến thứ i đang di chuyển theo chiều ngược lại. Tất cả các con kiến có tốc độ di chuyển bằng nhau & bằng 1 đơn vị đo độ dài trên giây.

Gọi t là thời điểm sớm nhất tính từ thời điểm quan sát mà tổng lượng mồi được tha về 2 tổ đạt ít nhất $\frac{1}{2}$ tổng lượng mồi của đàn kiến. Mr. Bean đếm được trong thời gian đó các con kiến gặp nhau đúng x lần, tính cả lần gặp nhau ở thời điểm t . Hỏi x bằng bao nhiêu?

- Input. Dòng 1: 2 số nguyên dương $n, l \in \mathbb{N}^*$, $1 \leq n \leq 5 \cdot 10^4$, $1 \leq l \leq 10^9$. Dòng 2, ..., $n + 1$: Dòng $i + 1$ ghi 3 số nguyên w_i, x_i, d_i , $1 \leq w_i \leq 10^3$, $d_i = \pm 1$, $0 \leq x_i \leq l$, các số x_i , $i = 1, 2, \dots, n$, đôi một phân biệt. Các số nguyên cách nhau 1 dấu cách.
- Output. 1 dòng duy nhất chứa số nguyên $x \in \mathbb{N}^*$ là số lần gặp nhau của các cặp kiến.
- Sample.

ant_meet.inp	ant_meet.out
3 5 1 1 1 2 2 -1 3 3 -1	2

Giải thích: Thời điểm 0.5, kiến 1 gặp kiến 2 ở tọa độ 1.5, kiến 1 đổi hướng thành -1 , kiến 2 đổi hướng thành 1 . Thời điểm 1, kiến 2 gặp kiến 3 ở tọa độ 2, kiến 2 đổi hướng thành -1 , kiến 3 đổi hướng thành 1 . Thời điểm 2: kiến 1 về đến tổ ở tọa độ 0. Thời điểm 3: kiến 2 về đến tổ ở tọa độ 0, lúc này lượng mồi đạt được ở 2 tổ là 3, bằng $\frac{1}{2}$ tổng lượng mồi của cả 3 kiến.

Bài toán 14 ([Tru23], 1., p. 20, HSG Lớp 12 Nam Định 2020–2021, Nearly perfect number – Số gần hoàn hảo). 1 số nguyên dương $a \in \mathbb{N}^*$ được gọi là số “gần hoàn hảo” nếu thỏa mãn điều kiện: $2a \leq k$ với k là tổng các ước số của a , e.g., 12 là 1 số “gần hoàn hảo” vì $2 \cdot 12 < 1 + 2 + 3 + 4 + 6 + 12$ ($24 < 28$).

- **Input.** Dòng đầu tiên chứa số nguyên dương $n \in \mathbb{N}^*$, $1 \leq n \leq 10^4$. n dòng tiếp theo, mỗi dòng là 1 số nguyên dương có giá trị $\leq 10^6$.
- **Output.** Dòng đầu tiên ghi số lượng số “gần hoàn hảo”. Các dòng tiếp theo, mỗi dòng ghi 1 số “gần hoàn hảo”, số gặp trước thì viết trước.
- **Sample.**

near_perfect_number.inp	near_perfect_number.out
5	2
8	12
16	6
12	
6	
7	

Bài toán 15 ([Tru23], 2., pp. 20–21, HSG Lớp 12 Nam Định 2020–2021, Special number – Số đặc biệt). Cho 1 dãy gồm n số nguyên a_1, a_2, \dots, a_n . Dãy \mathcal{S} đưa ra số đặc biệt trong dãy a . Số đặc biệt là số chỉ xuất hiện đúng 1 lần trong dãy số.

- **Input.** Dòng đầu tiên là số $n \in \mathbb{N}^*$, $1 \leq n \leq 10^6$. n dòng tiếp theo, dòng thứ i là số a_i , $|a_i| \leq 10^9$, $\forall i = 1, 2, \dots, n$.
- **Output.** Dòng đầu tiên ghi số lượng số đặc biệt. Các dòng tiếp theo, mỗi dòng ghi 1 số đặc biệt tính từ đầu dãy a .
- **Sample.**

special_number.inp	special_number.out
8	3
9	6
9	11
7	5
7	
6	
11	
9	
5	

Bài toán 16 ([Tru23], 3., p. 21, HSG Lớp 12 Nam Định 2020–2021, Game of gifts – Trò chơi tặng quà). 1 công ty có tổ chức trò chơi, tặng $n \in \mathbb{N}^*$ gói quà đã được chuẩn bị theo giá trị phần quà từ thấp đến cao, để tri ân cho n khách hàng. Công ty đó đã chuẩn bị 1 chiếc hộp đựng n mảnh giấy, mỗi mảnh giấy được bí mật ghi 1 mã hóa gồm nhiều ký tự số & chữ. Mỗi khách hàng được chọn 1 mảnh giấy trong chiếc hộp đó. Viết chương trình tặng quà từ thấp đến cao theo số lượng các ký tự số của mã hóa trong tờ giấy, nếu số lượng ký tự số trong mã hóa bằng nhau thì khách hàng chọn trước được tặng quà trước.

- **Input.** Dòng đầu tiên chứa số nguyên dương $n \in \mathbb{N}^*$, $1 \leq n \leq 10^4$. n dòng tiếp theo, mỗi dòng chứa 1 mã hóa không dài quá 255 ký tự tương ứng cho từng khách hàng.
- **Output.** Thứ tự tặng quà của trò chơi này cho n khách hàng trên.
- **Sample.**

game_gift.inp	game_gift.out
5	G2Chuc
N123456Cao	A89Dat
A89Dat	L512Ket
G2Chuc	E3689Qua
L512Ket	N123456Cao
E3689Qua	

Bài toán 17 ([Tru23], 4., pp. 21–22, HSG Lớp 12 Nam Định 2020–2021, Work – Công việc). Trong 1 dây chuyền làm việc của công ty có n công nhân làm n việc. Người ta đánh số cho công nhân từ 1 đến n theo thứ tự đứng trong dây chuyền. Thời gian hoàn thành 1 công việc của người thứ i là t_i phút. Mỗi người cần làm xong công việc của mình nhưng được quyền làm tối đa 2 việc. Vì thế họ có thể phối hợp với người đứng ngay trước mình cùng làm, nếu người thứ i & người thứ $i + 1$ phối hợp thì thời gian làm xong việc cho 2 người là p_i . Tìm phương án sao cho n công việc đều hoàn thành với thời gian ít nhất.

- **Input.** Dòng 1 ghi số n , $1 < n \leq 10^6$. Dòng 2 ghi thời gian làm xong việc của từng công nhân tương ứng trong dây chuyền t_1, t_2, \dots, t_n , $1 \leq t_i \leq 60$, $\forall i = 1, 2, \dots, n$. Dòng 3 ghi $n - 1$ số thời gian cùng làm tương ứng cho số cặp công nhân nếu phối hợp p_1, p_2, \dots, p_{n-1} , $1 \leq p_i \leq 100$, $\forall i = 1, 2, \dots, n$.
- **Output.** 1 số duy nhất ghi tổng thời gian hoàn thành công việc ít nhất của n công nhân.
- **Sample.**

work.inp	work.out
5	17
2 5 7 8 4	
3 9 10 10	

1 CSES Problem Set

1.1 Introductory Problems

Problem 1 (Weird Algorithm). Consider an algorithm that takes as input a positive integer $n \in \mathbb{N}^*$. If n is even, the algorithm divides it by 2, & if n is odd, the algorithm multiplies it by 3 & adds 1. The algorithm repeats this, until n is 1. E.g., the sequence for $n = 3$ is as follows: $3 \rightarrow 10 \rightarrow 5 \rightarrow 16 \rightarrow 8 \rightarrow 4 \rightarrow 2 \rightarrow 1$. Your task is to simulate the execution of the algorithm for a given value of n .

- **Input.** The only input line contains an integer $n \in \mathbb{Z}$.
- **Output.** Print a line that contains all values of n during the algorithm.
- **Constraints.** $1 \leq n \leq 10^6$.
- **Sample.**

weird_algorithm.inp	weird_algorithm.out
3	3 10 5 16 8 4 2 1

Source: [CSES Problem Set/weird algorithm](#), & [Laa20, Sect. 1.3, pp. 5–7].

Problem 2 (Missing Number). You are given all numbers between $1, 2, \dots, n$ except one. Your task is to find the missing number.

- **Input.** The 1st input line contains a positive integer $n \in \mathbb{N}^*$. The 2nd line contains $n - 1$ numbers. Each number is distinct & between 1 & n (inclusive).
- **Output.** Print the missing number.
- **Constraints.** $2 \leq n \leq 2 \cdot 10^5$.
- **Sample.**

missing_number.inp	missing_number.out
5	4
2 3 1 5	

Source: [CSES Problem Set/missing number](#).

Problem 3 (Repetitions). You are given a DNA sequence: a string consisting of characters A, C, G, & T. Your task is to find the longest repetition in the sequence. This is a maximum-length substring containing only 1 type of character.

- **Input.** The only input line contains a string of $n \in \mathbb{N}^*$ characters.
- **Output.** Print 1 integer: the length of the longest repetition.
- **Constraints.** $1 \leq n \leq 10^6$.

- Sample.

repetition.inp	repetition.out
ATTCGGGA	3

Source: [CSES problem Set/repetition](#).

Problem 4 (Nondecreasing Array). You are given an array of n integers. You want to modify the array so that it is non-decreasing, i.e., every element is at least as large as the previous element. On each move, you may increase the value of any element by 1. What is the minimum number of moves required?

- Input. The 1st input line contains an integer n : the size of the array. The 2nd line contains n integers x_1, x_2, \dots, x_n : the contents of the array.
- Output. Print the minimum number of moves.
- Constraints. $1 \leq n \leq 2 \cdot 10^5$, $1 \leq x_i \leq 10^9$.
- Sample.

nondecreasing_array.inp	nondecreasing_array.out
5 3 2 5 1 7	5

Source: [CSES problem Set/increasing array](#).

Problem 5 (Permutations). A permutation of integers $1, 2, \dots, n$ is called beautiful if there are no adjacent elements whose difference is 1. Given n , construct a beautiful permutation if such a permutation exists.

- Input. The only input line contains an integer n .
- Output. Print a beautiful permutation of integers $1, 2, \dots, n$. If there are several solutions, you may print any of them. If there are no solutions, print "no solution".
- Constraints. $1 \leq n \leq 10^6$.
- Sample.

permutations.inp	permutations.out
5	4 2 5 3 1
3	NO SOLUTION

Source: [CSES problem Set/permutations](#).

Problem 6 (Number Spiral). A number spiral is an infinite grid whose upper-left square has number 1. Here are the 1st 5 layers of the spiral:

1	2	9	10	25
4	3	8	11	24
5	6	7	12	23
16	15	14	13	22
17	18	19	20	21

Your task is to find out the number in row y & column x .

- Input. The 1st input line contains an integer t : the number of tests. After this, there are t lines, each containing integers y, x .
- Output. For each test, print the number in row y & column x .
- Constraints. $1 \leq t \leq 10^5$, $1 \leq x, y \leq 10^9$.
- Sample.

number_spiral.inp	number_spiral.out
3	8
2 3	1
1 1	15
4 2	

Source: [CSES problem Set/number spiral](#).

Problem 7 (2 Knights). Your task is to count for $k = 1, 2, \dots, n$ the number of ways 2 knights can be placed on a $k \times k$ chessboard so that they do not attack each other.

- Input. The only input line contains a positive integer $n \in \mathbb{N}^*$.
- Output. Print n integers: the results.
- Constraints. $1 \leq n \leq 10^4$.
- Sample.

knight.inp	knight.out
8	0 6 28 96 252 550 1056 1048

Source: [CSES problem Set/2 knights](#).

Problem 8 (2 Sets). Your task is to divide the numbers $1, 2, \dots, n$ into 2 sets of equal sum.

- Input. The only input line contains a positive integer $n \in \mathbb{N}^*$.
- Output. Print “yes”, if the division is possible, & “no” otherwise. After this, if the division is possible, print an example of how to create the sets. 1st, print the number of elements in the 1st set followed by the elements themselves in a separate line, & then, print the 2nd set in a similar way.
- Constraints. $1 \leq n \leq 10^6$.
- Sample.

set2.inp	set2.out
7	yes 4 1 2 4 7 3 3 5 6
6	no

Source: [CSES problem Set/2 sets](#).

Problem 9 (Bit String). Your task is to calculate the number of bit strings of length n . E.g., if $n = 3$, the correct answer is 8, because the possible bit strings are 000, 001, 010, 011, 100, 101, 110, & 111.

- Input. The only input line contains a positive integer $n \in \mathbb{N}^*$.
- Output. Print the result modulo $10^9 + 7$.
- Constraints. $1 \leq n \leq 10^6$.
- Sample.

bit_string.inp	bit_string.out
3	8

Source: [CSES problem Set/bit string](#).

Problem 10 (Trailing Zeros). Your task is to calculate the number of trailing zeros in the factorial $n!$, e.g., $20! = 2432902008176640000$ & it has 4 trailing zeroes.

- Input. The only input line contains a positive integer $n \in \mathbb{N}^*$.

- Output. Print the number of trailing zeros in $n!$.
- Constraints. $1 \leq n \leq 10^9$.
- Sample.

trailing_zero.inp	trailing_zero.out
20	4

Source: [CSES problem Set/trailing zeroes](#).

Problem 11 (Coin Piles). You have 2 coin piles containing a & b coins. On each move, you can either remove 1 coin from the left pile & 2 coins from the right pile, or 2 coins from the left pile & 1 coin from the right pile. Your task is to efficiently find out if you can empty both the piles.

- Input. The 1st input line has an integer t : the number of tests. After this, there are t lines, each of which has 2 integers a & b : the numbers of coins in the piles.
- Output. For each test, print “yes” if you can empty the piles & “no” otherwise.
- Constraints. $1 \leq t \leq 10^5$, $0 \leq a, b \leq 10^9$.
- Sample.

max_gift.inp	max_gift.out
3	yes
2 1	no
2 2	yes
3 3	

Source: [CSES problem Set/coin piles](#).

Problem 12 (Palindrome Reorder). Given a string, your task is to reorder its letters in such a way that it becomes a palindrome (i.e., it reads the same towards & backwards).

- Input. The only input line has a string of length n consisting of characters A-Z.
- Output. Print a palindrome consisting of the characters of the original string. You may print any valid solution. If there are no solutions, print “no solution”.
- Constraints. $1 \leq n \leq 10^6$.
- Sample.

palindrome_reorder.inp	palindrome_reorder.out
AAAACACBA	AACABACAA

Source: [CSES problem Set/palindrome reorder](#).

Problem 13 (Gray Code). A Gray code is a list of all 2^n bit strings of length n , where any 2 successive strings differ in exactly 1 bit (i.e., their Hamming distance is 1). Your task is to create a Gray code for a given length n .

- Input. The only input line has a positive integer $n \in \mathbb{N}^*$.
- Output. Print 2^n lines that describe the Gray code. You can print any valid solution.
- Constraints. $1 \leq n \leq 10^6$.
- Sample.

gray_code.inp	gray_code.out
2	00
	01
	11
	10

Source: [CSES problem Set/gray code](#).

Problem 14 (Tower of Hà Nội). The Tower of Hanoi game consists of 3 stacks (left, middle, & right) & n round disks of different sizes. Initially, the left stack has all the disks, in increasing order of size from top to bottom. The goal is to move all the disks to the right stack using the middle stack. On each move you can move the uppermost disk from a stack to another stack. In addition, it is not allowed to place a larger disk on a smaller disk. Your task is to find a solution that minimizes the number of moves.

- **Input.** The only input line has a positive integer $n \in \mathbb{N}^*$: the number of disks.
- **Output.** 1st print a positive integer $k \in \mathbb{N}^*$: the minimum number of moves. After this, print k lines that describe the moves. Each line has 2 integers $a, b \in \{1, 2, 3\}$: you move a disk from stack a to stack b .
- **Constraints.** $1 \leq n \leq 16$.
- **Sample.**

.inp	.out
2	3
	1 2
	1 3
	2 3

Source: [CSES problem Set/tower of Hà Nội](#).

Problem 15 (Creating Strings). Given a string, your task is to generate all different strings that can be created using its characters.

- **Input.** The only input line has a string of length $n \in \mathbb{N}^*$. Each character is between $a-z$.
- **Output.** 1st print a positive integer $k \in \mathbb{N}^*$: the number of strings. Then print k lines: the strings in alphabetical order.
- **Constraints.** $1 \leq n \leq 8$.
- **Sample.**

create_string.inp	create_string.out
aabac	20
	aaabc
	aaacb
	aabac
	aabca
	aacab
	aacba
	abaac
	abaca
	abcaa
	acaab
	acaba
	acbaa
	baaac
	baaca
	bacaa
	bcaaa
	caaab
	caaba
	cabaa
	cbaaa

Source: [CSES problem Set/creating strings](#).

Problem 16 (Apple Division). There are n apples with known weights. Your task is to divide the apples into 2 groups so that the difference between the weights of the groups is minimal.

- **Input.** The first input line has an integer $n \in \mathbb{N}^*$: the number of apples. The next line has n integers p_1, p_2, \dots, p_n : the weight of each apple.
- **Output.** Print 1 integer: the minimum difference between the weights of the groups.

- Constraints. $1 \leq n \leq 20$, $1 \leq p_i \leq 10^9$.
- Sample.

apple_division.inp	apple_division.out
5 3 2 7 4 1	1

Explanation: Group 1 has weights 2, 3, 4 (total weight 9), & group 2 has weights 1 & 7 (total weight 8).

Source: [CSES problem Set/apple division](#).

Problem 17 (Chessboard & Queens). Your task is to place 8 queens on a chessboard so that no 2 queens are attacking each other. As an additional challenge, each square is either free or reserved, & you can only place queens on the free squares. However, the reserved squares do not prevent queens from attacking each other. How many possible ways are there to place the queens?

- Input. The input has 8 lines, and each of them has 8 characters. Each square is either free (.) or reserved (*).
- Output. Print 1 integer: the number of ways you can place the queens.
- Sample. Input:

```

.....
.....
..*.....
.....
.....
.....**
...*....
.....

```

Output: 65

Source: [CSES problem Set/chessboard & queens](#).

Problem 18 (Digit Queries). Consider an infinite string that consists of all positive integers in increasing order:

123456789101112131415161718192021222324252627282930...

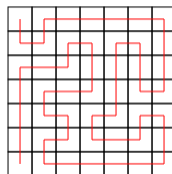
Your task is to process q queries of the form: what is the digit at position k in the string?

- Input. The 1st input line has a positive integer $q \in \mathbb{N}^+$: the number of queries. After this, there are q lines that describe the queries. Each line has an integer k : a 1-indexed position in the string.
- Output. For each query, print the corresponding digit.
- Constraints. $1 \leq q \leq 1000$, $1 \leq k \leq 10^{18}$.
- Sample.

digit_query.inp	digit_query.out
3	7
7	4
19	1
12	

Source: [CSES problem Set/digit queries](#).

Problem 19 (Grid Path). There are 88418 paths in a 7×7 grid from the upper-left square to the lower-left square. Each path corresponds to a 48-character description consisting of characters D (down), U (up), L (left), & R (right). E.g., the path



corresponds to the description DRURRRRRDDDLUULDDDLRRURDDLLLLLURULURRUULDLLDDDD. You are given a description of a path which may also contain characters ? (any direction). Your task is to calculate the number of paths that match the description.

- Input. The only input line has a 48-character string of characters ?, D, U, L, R.
- Output. Print 1 integer: the total number of paths.
- Sample.

grid_path.inp	grid_path.out
?????R?????U????????????????????LD????D?	201

1.2 Mathematics

Problem 20 (Josephus Queries). Consider a game where there are n children (numbered $1, 2, \dots, n$) in a circle. During the game, every second child is removed from the circle, until there are no children left. Your task is to process q queries of the form: “when there are n children, who is the k th child that will be removed?”

- Input. The 1st input line has a positive integer $q \in \mathbb{N}$: the number of queries. After this, there are q lines that describe the queries. Each line has 2 integers $n, k \in \mathbb{N}$: the number of children & the position of the child.
- Output. Print q integers: the answer for each query.
- Constraints. $1 \leq q \leq 10^5$, $1 \leq k \leq n \leq 10^9$.
- Sample.

Josephus_query.inp	Josephus_query.out
4	2
7 1	6
7 3	1
2 2	1107
1337 1313	

Tài liệu

- [Huy24] Nguyễn Xuân Huy. *Sáng Tạo Trong Thuật Toán & Lập Trình. Tập 1*. Tái bản lần thứ 10. Nhà Xuất Bản Thông Tin & Truyền Thông, 2024, p. 371.
- [Laa20] Antti Laaksonen. *Guide to Competitive Programming: Learning & Improving Algorithms Through Contests*. 2nd edition. Undergraduate Topics in Computer Science. Springer, 2020, pp. xv+309.
- [Tru23] Vương Thành Trung. *Tuyển Tập Đề Thi Học Sinh Giỏi Cấp Tỉnh Trung Học Phổ Thông Tin Học*. Tài liệu lưu hành nội bộ, 2023, p. 235.