



CHUYỂN CHỮ SỐ

Cho số nguyên dương a . Từ dãy chữ số thập phân của a , bạn được phép chọn đúng một chữ số rồi chuyển nó lên đầu hoặc xuống cuối dãy, giữ nguyên thứ tự các chữ số còn lại. Chú ý rằng bạn chỉ được thực hiện thao tác trên không quá 1 lần.

Yêu cầu: Tìm cách tạo ra biểu diễn thập phân của số lớn nhất có thể theo cách trên

Dữ liệu: Vào từ file văn bản DGMOVE.INP

☀ Dòng 1: Chứa số nguyên dương $T \leq 10^6$ là số test

☀ T dòng tiếp theo, mỗi dòng chứa một dãy chữ số liền nhau là biểu diễn thập phân của số a

Tổng độ dài các dãy chữ số của a trên tất cả T test không vượt quá 10^6

Kết quả: Ghi ra file văn bản DGMOVE.OUT T dòng, mỗi dòng ghi một dãy chữ số liền nhau là biểu diễn thập phân của số lớn nhất có thể tạo ra ứng với giá trị a trong dữ liệu.

Ví dụ

DGMOVE.INP	DGMOVE.OUT	Giải thích
5	9911	1991 → 9911
1991	4202	2024 → 4202
2024	992931	919293 → 992931
919293	8867256	8866725 → 8867256
8866725	33222110	
33222110		



TỔNG CÁC LŨY THỪA 2

Cho số nguyên dương n , hãy đếm số cách phân tích n thành tổng các lũy thừa của 2. Hai cách phân tích được gọi là khác nhau nếu chúng không phải hoán vị của nhau.

Ví dụ với $n = 8$ ta có 10 cách phân tích:

- 1) $8 = 1 + 1 + 1 + 1 + 1 + 1 + 1 + 1$
- 2) $8 = 2 + 1 + 1 + 1 + 1 + 1 + 1$
- 3) $8 = 2 + 2 + 1 + 1 + 1 + 1$
- 4) $8 = 2 + 2 + 2 + 1 + 1$
- 5) $8 = 2 + 2 + 2 + 2$
- 6) $8 = 4 + 1 + 1 + 1 + 1$
- 7) $8 = 4 + 2 + 1 + 1$
- 8) $8 = 4 + 2 + 2$
- 9) $8 = 4 + 4$
- 10) $8 = 8$

Dữ liệu: Vào từ file văn bản BINPAR.INP

☀ Dòng 1 chứa số nguyên dương $T \leq 100$ là số test

☀ T dòng tiếp theo, mỗi dòng chứa một số nguyên dương $n \leq 10^{18}$ ứng với một test

Kết quả: Ghi ra file văn bản BINPAR.OUT

Ứng với mỗi test, ghi ra trên một dòng một số nguyên duy nhất là số dư của phép chia kết quả tính được cho 1000000007 ($10^9 + 7$)

Ví dụ

BINPAR.INP	BINPAR.OUT
3	4
5	10
8	14
10	



THÊM CẠNH

Cho đồ thị vô hướng G , đếm số cặp đỉnh (u, v) không kề nhau ($u < v$) thỏa mãn: Nếu bổ sung thêm cạnh (u, v) ta được thêm đúng 1 chu trình đơn trên đồ thị. (Chu trình đơn là chu trình không đi qua một đỉnh 2 lần ngoại trừ nơi bắt đầu cũng là nơi kết thúc chu trình)

Dữ liệu: Vào từ file văn bản SIMPLEC.INP

- ☀ Dòng 1 chứa hai số nguyên dương $n \leq 10^5, m \leq 2 \cdot 10^5$ lần lượt là số đỉnh và số cạnh của đồ thị
- ☀ m dòng tiếp mỗi dòng chứa hai số nguyên dương là hai đỉnh đầu mút của một cạnh

Kết quả: Ghi ra file văn bản SIMPLEC.OUT một số nguyên duy nhất là kết quả tìm được

Ví dụ

SIMPLEC.INP	SIMPLEC.OUT
5 5	1
1 2	
1 3	
2 3	
1 4	
4 5	



DÂY NGOẶC ĐÚNG

Giáo sư X cho các bé trường SuperKids làm quen với thuật toán kiểm tra dãy ngoặc đúng. Một dãy ngoặc đúng là một xâu ký tự được định nghĩa đệ quy như sau:

- ✿ Xâu rỗng (không có ký tự nào) là một dãy ngoặc đúng
- ✿ Nếu X là một dãy ngoặc đúng thì (X) cũng là một dãy ngoặc đúng, ở đây (X) là xâu tạo thành bằng cách chèn thêm ký tự '(' vào đầu và ký tự ')' vào cuối xâu X
- ✿ Nếu Y và Z là hay dãy ngoặc đúng thì YZ cũng là một dãy ngoặc đúng, ở đây YZ là xâu tạo thành bằng cách nối xâu Z vào cuối xâu Y

Những xâu ký tự không thể xây dựng được theo quy tắc trên không phải là dãy ngoặc đúng.

Với một xâu ký tự chỉ gồm các ký tự $\in \{ '(', ')' \}$, ta gọi trọng số của xâu đó **số ký tự ít nhất cần chèn thêm** vào xâu ở các vị trí hợp lý để ta thu được một dãy ngoặc đúng. Ví dụ trọng số của xâu $((()$ cũng như xâu $()))(($ đều là 2 vì chúng cần chèn thêm ít nhất 2 ký tự mới trở thành dãy ngoặc đúng. Dưới đây là một trong những phương án chèn: $((() \rightarrow ()()() ; ())((\rightarrow (())(())$.

Cho xâu ký tự $S = S_1S_2 \dots S_n$ chỉ gồm các ký tự $\in \{ '(', ')' \}$ (chú ý là các ký tự trong xâu đánh số từ 1 trở đi), người ta thực hiện tuần tự m lệnh (đánh số từ 1 tới m) thuộc một trong ba loại:

- ✿ $C\ i$: Đảo ký tự S_i từ ký tự '(' thành ký tự ')' và ngược lại ($1 \leq i \leq n$)
- ✿ $Q\ i\ j$: Cho biết trọng số của xâu con gồm các ký tự liên tiếp $S_iS_{i+1} \dots S_j$ ($1 \leq i \leq j \leq n$)
- ✿ $U\ k$: Phục hồi lại xâu S như tình trạng trước khi thực hiện lệnh thứ k (k là một số nguyên dương nhỏ hơn hoặc bằng chỉ số lệnh hiện tại)

Yêu cầu: Hãy mô phỏng quá trình thực hiện lệnh và cho biết câu trả lời cho các lệnh Q

Dữ liệu: Vào từ file văn bản FIXSTR.INP

- ✿ Dòng 1 chứa xâu S gồm không quá 10^6 ký tự $\in \{ '(', ')' \}$
- ✿ Dòng 2 chứa số nguyên dương $m \leq 10^6$ là số lệnh
- ✿ m dòng tiếp theo chứa thông tin về một lệnh theo mô tả trên.

Kết quả: Ghi ra file văn bản FIXSTR.OUT, với mỗi lệnh Q , ghi ra một số nguyên duy nhất trên một dòng là câu trả lời cho lệnh đó.

Ví dụ

FIXSTR.INP	FIXSTR.OUT	Giải thích
((()))	0	Xâu S sau mỗi lệnh
10	2	1:)((())
C 1	0	2:)(())
Q 2 5	3	3:)((()()
C 5		4:)(())()
Q 1 6		5:)((())
U 3		6:)((())
C 4		7:)(())()
Q 3 6		8:)((()()
U 5		9: ((()()
C 1		10: (((())
Q 1 5		



GIẢI NGHỊCH THỂ

Cho dãy số nguyên $A = (a_1, a_2, \dots, a_n)$ trong đó $a_i \in \{0,1\}$.

Số nghịch thể của dãy A được định nghĩa là số cặp chỉ số (i, j) thỏa mãn hai điều kiện:

- ☀ $i < j$
- ☀ $a_i > a_j$

Xét phép đảo tại vị trí i : Biến a_i thành $1 - a_i$, tức là phép đảo tại vị trí i sẽ biến a_i đang từ 0 thành 1 và đang từ 1 thành 0.

Yêu cầu: Bạn cần trả lời q truy vấn, mỗi truy vấn cho bởi một số k : Cho biết nếu được thực hiện không quá k phép đảo trên dãy A **ban đầu**, thì số nghịch thể của dãy tạo thành tối thiểu là bao nhiêu.

Dữ liệu: Vào từ file văn bản DNT.INP

- ☀ Dòng 1 chứa hai số nguyên dương $n \leq 10^6, q \leq 10$
- ☀ Dòng 2 chứa n số nguyên a_1, a_2, \dots, a_n cách nhau bởi dấu cách ($\forall i: a_i \in \{0,1\}$)
- ☀ Dòng 3 ghi q số k_1, k_2, \dots, k_q mỗi số ứng với một truy vấn ($0 \leq k_i \leq n$)

Kết quả: Ghi ra file văn bản DNT.OUT q dòng, mỗi dòng i ghi một số nguyên duy nhất là số nghịch thể của dãy số tạo thành sau không quá k_i phép đảo

Chú ý: Các truy vấn là độc lập, tức là với mỗi truy vấn k_i bạn có thể thực hiện không quá k_i phép đảo trên dãy A cho từ input, không phải dãy A đã biến đổi từ các truy vấn trước đó.

Ví dụ

DNT.INP	DNT.OUT	Giải thích
3 2	0	1 0 1 \rightarrow 1 1 1
1 0 1	0	1 0 1 \rightarrow 0 0 0
1 2		
4 2	2	1 1 0 0 \rightarrow 0 1 0 0
1 1 0 0	0	1 1 0 0 \rightarrow 1 1 1 1
1 2		



ĐẢO CHỮ

Giáo sư X thiết kế một khẩu hiệu cực lớn dưới dạng xâu ký tự $S = s_0s_1 \dots s_{n-1}$ và thuê sơn lên các toa của đoàn tàu xuyên Việt theo đúng thứ tự từ toa đầu tiên tới toa cuối cùng, mỗi toa một ký tự. Tuy nhiên sau khi sơn thì các toa lại được nhân viên đường sắt sắp xếp lại theo thứ tự lộn xộn khiến cho các ký tự trong biểu ngữ có thể bị hoán vị, biểu ngữ trở thành xâu $A = a_0a_1 \dots a_{n-1}$.

Rất may là giáo sư X thuê được một đầu máy và một đường ray phụ, với những thiết bị này, ông có thể thực hiện những thao tác: kéo một toa sang đường ray phụ rồi chuyển lại về đầu hoặc cuối dãy toa tàu tùy theo tính toán của ông.

Yêu cầu: Hãy giúp giáo sư X khôi phục lại thứ tự của biểu ngữ bằng ít thao tác nhất.

Dữ liệu: Vào từ file văn bản ANAGRAM.INP

☀ Dòng 1 chứa xâu S

☀ Dòng 2 chứa xâu A

Xâu S chỉ chứa các chữ cái thường, độ dài không quá 5000, xâu A là hoán vị của các ký tự trong xâu S

Kết quả: Ghi ra file văn bản ANAGRAM.OUT một số nguyên duy nhất là số thao tác ít nhất cần thực hiện

Ví dụ

ANAGRAM.INP	ANAGRAM.OUT
goldenland oldengland	1
twelveplusone elevenplustwo	4
dormitory dirtyroom	6

Giải thích

olden**g**land → **g**oldenland

elevenplust**w**o → **w**elevenplusto

welevenplust**o** → **t**welevenplusto

twelevenplust**o** → twelevenplust**o**

twelevenplust**o** → twelevenplust**o**