Bài A. NUM13

File dữ liệu vào: `stdin`
File kết quả: `stdout`
Hạn chế thời gian: 2 giây

Cho số nguyên dương n. Hãy đếm xem có bao nhiêu số nguyên dương x chia hết cho 13; sao cho ở dạng biểu diễn thập phân, x có đúng n chữ số, đồng thời x không chứa chữ số 1 và 3.

Dữ liệu vào

Gồm một số nguyên dương n ($1 \leq n \leq 10^5$)

Kết quả

Ghi số lượng số nguyên dương x tìm được, sau khi chia lấy dư cho 1000000007

Ví dụ

stdin	stdout
1	0
2	4

Giải thích

Các số thỏa mãn test VD 2 là 39, 52, 65, 78

Hạn chế

- Có 20% số test với $n \leq 6$

Bài B. NUM17

File dữ liệu vào:	`stdin`
File kết quả:	`stdout`
Hạn chế thời gian:	2 giây

Cho hai số nguyên dương L, R. Hãy đếm xem có bao nhiêu số nguyên dương x thuộc đoạn $[L, R]$ hết cho 17; sao cho ở dạng biểu diễn thập phân, x không chứa hai chữ số bằng nhau nằm kề nhau

Dữ liệu vào

Gồm hai dòng chứa hai số nguyên dương L, R ($1 \leq L \leq R \leq 10^{10000}$)

Kết quả

Ghi số lượng số nguyên dương x tìm được, sau khi chia lấy dư cho 1000000007

Ví dụ

stdin	stdout
1 100	5

Giải thích

Các số thỏa mãn là 17, 34, 51, 68, 85

Hạn chế

- Có 20% số test với $R \leq 10^6$

Bài C. PLDNUM

File dữ liệu vào:	`stdin`
File kết quả:	`stdout`
Hạn chế thời gian:	2 giây

Cho số nguyên dương R. Hãy đếm xem có bao nhiêu số nguyên dương $x \leq R$; sao cho ở dạng biểu diễn thập phân, x là một xâu đối xứng (khi đọc từ trái sang phải cũng giống khi đọc từ phải sang trái) và hai chữ số cạnh nhau chênh lệch nhau ít nhất 5 đơn vị?

Dữ liệu vào

Chứa số nguyên dương R ($1 \leq R \leq 10^{100000}$)

Kết quả

Ghi số lượng số nguyên dương x tìm được, sau khi chia lấy dư cho 1000000007

Ví dụ

stdin	stdout
700	22

Giải thích

Các số thỏa mãn là 1, 2, 3, 4, 5, 6, 7, 8, 9, 161, 171, 181, 191, 272, 282, 292, 383, 393, 494, 505, 606, 616

Hạn chế

- Có 20% số test với $R \leq 10^6$
- Có 30% số test với $R \leq 10^{1000}$

Bài D. BK1

File dữ liệu vào:	`stdin`
File kết quả:	`stdout`
Hạn chế thời gian:	1 giây

Các dấu ngoặc xuất hiện rất nhiều trong các biểu thức toán học để thể hiện thứ tự tính toán. Giờ đây ta bỏ hết các hạng tử toán tử đi, chỉ giữ lại các dấu ngoặc, biểu thức mà ta thu được gọi là một dãy ngoặc đúng. Cụ thể hơn:

- Xâu rỗng là biểu thức ngoặc đúng bậc 0
- Nếu A là biểu thức ngoặc đúng bậc k thì (A) là dãy ngoặc đúng bậc $k+1$
- Nếu A là biểu thức ngoặc đúng bậc a và B là biểu thức ngoặc đúng bậc b thì AB là biểu thức ngoặc đúng bậc $\max(a, b)$

Cho một xâu S chỉ chứa các ký tự $'('$, $')'$ và một số k. Hùng muốn đánh dấu một số vị trí trên xâu này sao cho khi xóa các vị trí bị đánh dấu đó đi Hùng thu được dãy ngoặc đúng bậc k. Hai cách đánh dấu được coi là khác nhau nếu tồn tại một vị trí được đánh dấu trong cách này nhưng không được đánh dấu trong cách kia

Dữ liệu vào

- Dòng đầu tiên chứa số tự nhiên k
- Dòng tiếp theo chứa xâu S

Kết quả

Ghi số cách xóa tìm được, chỉ cần in ra phần dư khi chia cho $10^9 + 7$

Ví dụ

stdin	stdout
1 ())	2

Hạn chế

- $1 \le k \le |S| \le 5000$, $|S|$ chẵn
- Subtask 1: $|S| \le 10$
- Subtask 2: Ràng buộc gốc

Bài E. BK2

File dữ liệu vào: `stdin`
File kết quả: `stdout`
Hạn chế thời gian: 1 giây

Các dấu ngoặc xuất hiện rất nhiều trong các biểu thức toán học để thể hiện thứ tự tính toán. Giờ đây ta bỏ hết các hạng tử toán tử đi, chỉ giữ lại các dấu ngoặc, biểu thức mà ta thu được gọi là một dãy ngoặc đúng. Cụ thể hơn:

- Xâu rỗng là biểu thức ngoặc đúng bậc 0
- Nếu A là biểu thức ngoặc đúng bậc k thì (A) là dãy ngoặc đúng bậc $k+1$
- Nếu A là biểu thức ngoặc đúng bậc a và B là biểu thức ngoặc đúng bậc b thì AB là biểu thức ngoặc đúng bậc $\max(a, b)$

Cho một xâu S chỉ chứa các ký tự $'('$, $')'$ và một số k. Hùng muốn xóa đi một số ký tự trên S và giữ nguyên thứ tự của các ký tự còn lại sao cho xâu thu được là dãy ngoặc đúng bậc k. Hãy đếm số cách để Hùng làm điều đó, hai cách xóa được coi là khác nhau nếu xâu thu được của chúng là khác nhau

Dữ liệu vào

- Dòng đầu tiên chứa số tự nhiên k
- Dòng tiếp theo chứa xâu S

Kết quả

Ghi số cách xóa tìm được, chỉ cần in ra phần dư khi chia cho $10^9 + 7$

Ví dụ

stdin	stdout
1 ())	1

Hạn chế

- $1 \leq k \leq |S| \leq 5000$, $|S|$ chẵn
- Subtask 1: $|S| \leq 10$
- Subtask 2: Ràng buộc gốc

Bài F. NUM19

File dữ liệu vào:	`stdin`
File kết quả:	`stdout`
Hạn chế thời gian:	2 giây

Cho hai số nguyên dương L, R. Hãy đếm xem có bao nhiêu số nguyên dương x thuộc đoạn $[L, R]$ hết cho 19; sao cho ở dạng biểu diễn thập phân, x không chứa hai chữ số nào có tổng chia hết cho 3

Dữ liệu vào

- Dòng đầu chứa số nguyên dương T là số lượng testcase ($1 \leq T \leq 10^5$)
- Mỗi test được mô tả trên 2 dòng là L và R ($1 \leq L \leq R \leq 10^{10000}$)

Tổng độ dài của tất cả các số R trong T testcase không vượt quá 10^5

Kết quả

Với mỗi testcase, ghi trên một dòng số lượng số nguyên dương x tìm được, sau khi chia lấy dư cho 1000000007

Ví dụ

stdin	stdout
2 1 100 101 200	4 2

Giải thích

Các số thỏa mãn testcase 1 là: 19, 38, 76, 95

Hạn chế

- Có 20% số test với $R \leq 10^6$
- Có 30% số test với tổng độ dài của tất cả các số R trong T testcase không vượt quá 10^3

Bài G. ENCODE

File dữ liệu vào:	`stdin`
File kết quả:	`stdout`
Hạn chế thời gian:	2 giây

Trọng số của một số tự nhiên x được tính như sau:

- Xét biểu diễn thập phân của x (không chứa các số 0 đứng đầu). Chia nó thành ít đoạn con liên tiếp nhất sao cho trong mỗi đoạn con, các chữ số đều giống nhau.
- Với mỗi đoạn con, gọi hàng to nhất là e và chữ số ở hàng thứ e là i. Trọng số của đoạn này sẽ là $i \times 10^e$.
- Trọng số của x là tổng trọng số của tất cả các đoạn con.

Ví dụ: 388822442 có thể được chia ra thành các đoạn con: "3", "888", "22", "44", "2", trong đó $e = 7$ với đoạn con "888" và $e = 4$ với đoạn con "22". Trọng số của 388822442 là $3 \times 10^8 + 8 \times 10^7 + 2 \times 10^4 + 4 \times 10^2 + 2 \times 10^0$.

Yêu cầu: Cho hai số tự nhiên L, R; hãy tính tổng trọng số của tất cả các số tự nhiên x thuộc đoạn $[L, R]$. Do kết quả có thể rất lớn, hãy in ra đáp số với phần dư khi chia cho $10^9 + 7$.

Dữ liệu vào

- Dòng đầu chứa số nguyên dương T là số lượng testcase ($1 \leq T \leq 100000$)
- Mỗi testcase được cho trên hai dòng, là L và $R (L \leq R \leq 10^{100000})$

Tổng độ dài của tất cả các số R trong T testcase không vượt quá 10^6

Kết quả

Ghi T dòng là kết quả cho T testcase

Ví dụ

stdin	stdout
3 9 97 8 12 1 8	4681 49 36

Hạn chế

- Có 20% số test với $R \leq 10^6$
- Có 30% số test với tổng độ dài của tất cả các số R trong T testcase không vượt quá 10^3

Bài H. BR2

File dữ liệu vào:	`stdin`
File kết quả:	`stdout`
Hạn chế thời gian:	1 giây

Các dấu ngoặc xuất hiện rất nhiều trong các biểu thức toán học để thể hiện thứ tự tính toán. Giờ đây ta bỏ hết các hạng tử toán tử đi, chỉ giữ lại các dấu ngoặc, biểu thức mà ta thu được gọi là một dãy ngoặc đúng. Cụ thể hơn:

- Xâu rỗng là biểu thức ngoặc đúng
- Nếu A là biểu thức ngoặc đúng thì (A) cũng là dãy ngoặc đúng
- Nếu A và B là biểu các thức ngoặc đúng thì AB cũng là biểu thức ngoặc đúng

Cho S là một biểu thức ngoặc đúng có độ dài n. Tiến hành sắp xếp tất cả các dãy ngoặc đúng độ dài n theo thứ tự từ điển (với '(' < ')'), sau đó đánh số thứ tự cho các xâu bắt đầu từ 1. Tìm số được đánh cho S. Bạn cần trả lời nhiều truy vấn như vậy, độ dài xâu S ở các truy vấn có thể khác nhau

Dữ liệu vào

- Dòng đầu tiên chứa Q là số lượng truy vấn
- Q dòng tiếp theo mỗi dòng chứa một xâu S

Kết quả

- Gồm Q dòng. Mỗi dòng chứa số nguyên dương là thứ tự của S với test tương ứng, chỉ cần in ra phần dư khi chia cho $10^9 + 7$

Ví dụ

stdin	stdout
3 ()()() () ()()	5 1 2

Hạn chế

- $1 \leq |S|, Q \leq 10^3$, $|S|$ chẵn
- Subtask 1: Tất cả các xâu S có cùng độ dài
- Subtask 2: Ràng buộc gốc

Bài I. POW3

File dữ liệu vào:	`stdin`
File kết quả:	`stdout`
Hạn chế thời gian:	2 giây

Cho ba hai số nguyên dương L, R, k. Hãy đếm xem có bao nhiêu số nguyên dương x thuộc đoạn $[L, R]$; sao cho x có thể phân tích thành tổng của không quá k số là lũy thừa của 3

Dữ liệu vào

- Dòng đầu chứa số nguyên dương T là số lượng testcase ($1 \leq T \leq 10^5$)
- Mỗi test được mô tả trên 3 dòng là L, R và k ($1 \leq L \leq R \leq 10^{1000}$, $1 \leq k \leq 1000$)

Tổng độ dài của tất cả các số R trong T testcase không vượt quá 10^5

Kết quả

Với mỗi testcase, ghi trên một dòng số lượng số nguyên dương x tìm được, sau khi chia lấy dư cho 1000000007

Ví dụ

stdin	stdout
2 1 100 3 101 200 2	40 2

Giải thích

Các số thỏa mãn testcase 2 là: 108, 162

Hạn chế

- Có 20% số test với $R \leq 10^6$
- Có 30% số test với tổng độ dài của tất cả các số R trong T testcase không vượt quá 10^3

Bài J. FNUMBER

File dữ liệu vào:	`stdin`
File kết quả:	`stdout`
Hạn chế thời gian:	1 giây

Số nguyên dương n được gọi là FNUMBER nếu n chia hết cho 3 nhưng không chia hết cho 9 và ở dạng biểu diễn thập phân của n không có 2 chữ số nào có tổng chia hết cho 5. VD 240 là một FNUMBER, 231 không là một FNUMBER

Cho trước 2 số nguyên dương A, B. Đếm số lượng FNUMBER thuộc $[A; B]$. Do kết quả có thể rất lớn, chỉ cần in ra phần dư khi chia cho $10^9 + 7$

Dữ liệu vào

- Dòng 1 chứa T là số lượng testcase
- T dòng tiếp theo, mỗi dòng chứa 2 số A B

Kết quả

- Gồm T ghi kết quả tương ứng cho T testcase

Ví dụ

stdin	stdout
3 100 110 5 2345 0 100000	1 209 3616
1 0 1000000000000000000	698089379

Hạn chế

- Subtask 0: $T \leq 10^6$, $0 \leq A \leq B \leq 10^5$
- Subtask 1: $T = 1$, $0 \leq A \leq B \leq 10^{18}$
- Subtask 2: $T \leq 100$, $0 \leq A \leq B \leq 10^{10000}$

Bài K. BRCNT1

File dữ liệu vào:	`stdin`
File kết quả:	`stdout`
Hạn chế thời gian:	1 giây

Các dấu ngoặc xuất hiện rất nhiều trong các biểu thức toán học để thể hiện thứ tự tính toán. Giờ đây ta bỏ hết các hạng tử toán tử đi, chỉ giữ lại các dấu ngoặc, biểu thức mà ta thu được gọi là một dãy ngoặc đúng. Cụ thể hơn:

- Xâu rỗng là biểu thức ngoặc đúng bậc 0
- Nếu A là biểu thức ngoặc đúng bậc k thì (A), $[A]$, $\{A\}$ đều là các biểu thức ngoặc đúng bậc $k+1$
- Nếu A là biểu thức ngoặc đúng bậc a, B là biểu thức ngoặc đúng bậc b thì AB là biểu thức ngoặc đúng bậc $\max(a, b)$

Cho hai số nguyên dương n và k. Ta tiến hành sắp xếp các biểu thức ngoặc đúng có độ dài n và bậc k theo thứ tự từ điển, với $'(' < '[' < '\{' < ')' < ']' < '\}'$. Yêu cầu:

- Cho S là một biểu thức ngoặc đúng có độ dài n, bậc k. Tìm thứ tự của S ở trong dãy đã sắp xếp ở trên
- Cho số nguyên dương p. Đưa ra dãy ngoặc thứ p ở trong dãy đã sắp xếp ở trên

Dữ liệu vào

- Dòng đầu tiên chứa hai số nguyên dương n k
- Dòng thứ hai chứa xâu S
- Dòng thứ ba chứa số nguyên dương p

Kết quả

- Dòng đầu chứa số nguyên dương là thứ tự của S, dữ liệu đảm bảo số này không quá 10^{18}
- Dòng thứ hai chứa biểu thức ngoặc thứ p

Ví dụ

stdin	stdout
4 2 ([]) 4	2 [()]
6 2 ()[{}] 20	24 ()([])

Hạn chế

- $1 \le k \le n \le 10^3$, n chẵn. $1 \le p \le 10^{18}$
- Subtask 0: $n \le 10$
- Subtask 1: S là xâu có thứ tự 1
- Subtask 2: $p = 1$
- Subtask 3: Ràng buộc gốc

Bài L. SUM4

File dữ liệu vào:	`stdin`
File kết quả:	`stdout`
Hạn chế thời gian:	1 giây

Đếm số nghiệm nguyên của hệ (ở đây x là biến):

$$x_1 + x_2 + x_3 + x_4 = N$$
$$L_i \leq x_i \leq H_i, \forall i \in \{1, 2, 3, 4\}$$

Dữ liệu vào

- Dòng đầu tiên chứa Q là số lượng testcase
- Q dòng tiếp theo, mỗi dòng chứa: L_1 H_1 L_2 H_2 L_3 H_3 L_4 H_4 N

Kết quả

- Q dòng là kết quả Q bài toán sau khi chia lấy dư cho 1000000007

Ví dụ

stdin	stdout
1 -5 5 -5 5 -5 5 -5 5 10	286

Hạn chế

- $-10^9 \leq L_i \leq H_i \leq 10^9$; $-10^9 \leq N \leq 10^9$; $1 \leq Q \leq 1000$
- Subtask 1: $|L_i|, |H_i| \leq 100$
- Subtask 2: $|L_i|, |H_i| \leq 10^5$
- Subtask 3: $Q = 1$
- Subtask 4: Ràng buộc gốc

Bài M. BRK

File dữ liệu vào:	`stdin`
File kết quả:	`stdout`
Hạn chế thời gian:	1 giây

Cho xâu S độ dài n chỉ chứa các ký tự thuộc tập {'{', '[', '(', '}', ']', ')' ,'?'}. Đếm số cách thay các dấu '?' bằng các ký tự khác để thu được dãy ngoặc đúng có bậc bằng k. Dãy ngoặc đúng bậc k được định nghĩa như sau:

- Xâu rỗng là dãy ngoặc đúng bậc 0
- Nếu A là dãy ngoặc đúng bậc a, B là dãy ngoặc đúng bậc b thì: {A}, (A), [A] là các dãy ngoặc đúng bậc $a + 1$. AB là dãy ngoặc đúng bậc $\max(a, b)$

Dữ liệu vào

- Dòng đầu chứa hai số nguyên dương: n k
- Dòng thứ hai chứa một xâu có độ dài n, gồm các ký tự thuộc tập {'{', '[', '(', '}', ']', ')', '?'}

Kết quả

Ghi số cách thay thế, sau khi chia lấy dư cho $10^9 + 7$

Ví dụ

stdin	stdout
8 2 ?(??]?{}	3
8 3 ?(??]?{}	0

Hạn chế

- $1 \leq n \leq 2000$, $1 \leq k \leq 5$