

## OTHER NUMBER THEORIES (C PROGRAMMING)

### IN RA TAM GIÁC PASCAL CÁC CHỮ SỐ

Tam giác Pascal là tam giác có công thức tính giá trị một vị trí bất kỳ như sau

$$(n,k) = n!/k!(n-k)!$$

Trong đó: n là hàng và k là cột.

Người ta thường tận dụng hàm đệ quy để điền các giá trị cho tam giác Pascal.

Viết chương trình sử dụng đệ quy nhập vào số nguyên dương n không quá 10 và in ra tam giác Pascal tương ứng (lưu ý mỗi phần tử cách nhau 1 khoảng trắng)

Ví dụ:

Input	Output
5	1 1 1 1 2 1 1 3 3 1 1 4 6 4 1

Giới hạn thời gian: 2s

Giới hạn bộ nhớ: 65536 Kb

-----

ĐẾM CÁC SỐ GIẢM TRONG ĐOẠN [A;B]

## OTHER NUMBER THEORIES (C PROGRAMMING)

Một số gọi là số giảm nếu các chữ số từ trái qua phải tạo thành dãy giảm (tức là chữ số phía sau nhỏ hơn chữ số ngay trước). Viết chương trình đếm các số giảm có trong đoạn  $[a,b]$ .

### Input

- Dòng đầu ghi số bộ test
- Mỗi bộ test ghi 2 số  $a,b$  ( $1 < a < b < 108$ )

### Output

- Với mỗi bộ test, ghi ra số lượng các số giảm đếm được.

### Ví dụ

Input	Output
1	45
10 100	

Giới hạn thời gian: 1s

Giới hạn bộ nhớ: 65536 Kb

---

LIỆT KÊ CÁC SỐ VỪA NGUYÊN TỐ VỪA THUẬN NGHỊCH – IN MỖI DÒNG  
10 SỐ

## OTHER NUMBER THEORIES (C PROGRAMMING)

Viết chương trình liệt kê các số vừa nguyên tố, vừa thuận nghịch trong đoạn  $[a,b]$  với  $a,b$  là các số nguyên dương thỏa mãn  $10 < a < b < 106$ .

Input

Dòng đầu ghi số bộ test. Mỗi bộ test là một cặp số  $a,b$ .

Output

Với mỗi bộ test, ghi lần lượt các số thỏa mãn, mỗi số cách nhau một khoảng trống, mỗi dòng ghi đúng 10 số. Khi hết một test thì bỏ trống một dòng trước khi ghi kết quả test tiếp theo.

Ví dụ

Input	Output
2	101 131 151 181 191 313 353 373 383 727
20 1234	757 787 797 919 929
123 140	131

Giới hạn thời gian: 3s

Giới hạn bộ nhớ: 65536 Kb

---

LIỆT KÊ CÁC SỐ CÓ SỐ LƯỢNG CHỮ SỐ CHẴN BẰNG SỐ LƯỢNG CHỮ SỐ  
LẺ - IN MỖI DÒNG 10 SỐ

OTHER NUMBER THEORIES (C PROGRAMMING)

Cho số nguyên dương  $n$  không quá 9 chữ số, liệt kê các số có  $n$  chữ số thỏa mãn tính chất trong nó số chữ số lẻ bằng số chữ số chẵn.

Input

Dòng đầu ghi số bộ test. Mỗi bộ test là một số nguyên dương có  $n$  không quá 9 chữ số

Output

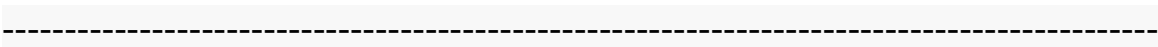
Với mỗi bộ test, ghi lần lượt các số thỏa mãn, mỗi số cách nhau một khoảng trống, mỗi dòng ghi đúng 10 số. Khi hết một test thì bỏ trống một dòng trước khi ghi kết quả test tiếp theo.

Ví dụ

Input	Output
1	10 12 14 16 18 21 23 25 27 29
2	30 32 34 36 38 41 43 45 47 49
	50 52 54 56 58 61 63 65 67 69
	70 72 74 76 78 81 83 85 87 89
	90 92 94 96 98

Giới hạn thời gian: 3s

Giới hạn bộ nhớ: 65536 Kb



## OTHER NUMBER THEORIES (C PROGRAMMING)

### LIỆT KÊ SỐ KHÔNG GIẢM DỄ BỊ TLE

Một số gọi là số không giảm nếu các chữ số từ trái qua phải tạo thành dãy không giảm (tức là không có chữ số nào phía sau nhỏ hơn chữ số ngay trước nó). Viết chương trình liệt kê các số không giảm có N chữ số ( $1 < N < 7$ ).

#### Input

- Dòng đầu ghi số bộ test
- Mỗi bộ test ghi số N

#### Output

- Với mỗi bộ test, ghi ra lần lượt các số không giảm có N chữ số, các số cách nhau một khoảng trống.
- Hết một bộ test thì xuống dòng.

#### Ví dụ

Input	Output
1	11 12 13 14 15 16 17 18 19 22 23 24 25 26 27 28 29 33 34 35 36 37 38 39
2	44 45 46 47 48 49 55 56 57 58 59 66 67 68 69 77 78 79 88 89 99

*Ghi chú: Kết quả của một test được viết trên một dòng. Trong bảng ví dụ trên do kích thước màn hình nên chia thành nhiều dòng cho dễ quan sát.*

Giới hạn thời gian: 1s

Giới hạn bộ nhớ: 65536 Kb

-----

## OTHER NUMBER THEORIES (C PROGRAMMING)

### ĐẾM CÁC SỐ MÀ LÀ SỐ THUẬN NGHỊCH VÀ TỔNG CHỮ SỐ CHIA HẾT 10

Một số được coi là đẹp nếu nó có tính chất thuận nghịch và tổng chữ số chia hết cho 10. Bài toán đặt ra là cho trước số chữ số. Hãy đếm xem có bao nhiêu số đẹp với số chữ số như vậy.

Dữ liệu vào: Dòng đầu tiên ghi số bộ test. Mỗi bộ test viết trên một dòng số chữ số tương ứng cần kiểm tra (lớn hơn 1 và nhỏ hơn 10).

Kết quả: Mỗi bộ test viết ra số lượng số đẹp tương ứng.

Ví dụ:

Input	Output
2	1
2	90
5	

Giới hạn thời gian: 3s

Giới hạn bộ nhớ: 65536 Kb

---

### SỐ MAY MẮN 4 VÀ 7

Chữ số 4 và 7 được coi là các chữ số may mắn.

Cho số nguyên dương N có không quá 18 chữ số. Hãy đếm xem chữ số 4 cộng với chữ số 7 trong N có phải bằng chính 4 hoặc 7 hay không?

## OTHER NUMBER THEORIES (C PROGRAMMING)

Input

Chỉ có số N

Output

Ghi ra YES hoặc NO tùy thuộc vào kết quả.

Ví dụ

Input	Output
40047	NO
7747774	YES
100000000000	NO

Giới hạn thời gian: 1s

Giới hạn bộ nhớ: 65536 Kb

---

NHẬP VÀO SỐ N, TÍNH KẾT QUẢ CỦA K CHỮ SỐ ĐẦU TIÊN CỦA N CHIA CHO CHÍNH K

Nhập vào một số N, tính kết quả của K chữ số đầu tiên của N chia cho chính K.

## OTHER NUMBER THEORIES (C PROGRAMMING)

Dữ liệu vào: Chỉ có 1 dòng ghi số nguyên dương  $N$  từ 3 đến 18 chữ số, một khoảng trống sau đó đến số nguyên dương  $k$ , đảm bảo nhỏ hơn số chữ số của  $N$

Kết quả: Ghi ra kết quả được tính trên 1 dòng

Ví dụ:

Input	Output
123456789012 4	308

Giải thích ví dụ: 4 số đầu tiên của  $N$  là 1234, kết quả bằng  $1234 / 4 = 308$  (chỉ lấy phần nguyên)

Giới hạn thời gian: 3s

Giới hạn bộ nhớ: 65536 Kb

-----

### DIGITAL ROOT

Cho số nguyên dương không âm  $N$ , ở mỗi thao tác bạn thực hiện tính tổng các chữ số của  $N$  sau đó gán lại cho  $N$ , thao tác này được thực hiện cho tới khi  $N$  chỉ còn 1 chữ số.

**Ví dụ**  $N = 278 \rightarrow 17 \rightarrow 8$ , vậy ta có dạng rút gọn của 278 là 8. Nhiệm vụ của bạn là tìm dạng rút gọn của 1 số nguyên không âm  $N$  cho trước

Đầu vào

1 dòng chứa số  $N$

**Giới hạn**



## OTHER NUMBER THEORIES (C PROGRAMMING)

$0 \leq N \leq 10^{18}$

Đầu ra

In ra dạng rút gọn của **N**

Input 01

999991020

Output 01

3