

# Problema A

## Palitos

Un niño quiere construir un número de hasta 50 dígitos decimales (0-9) a base de palitos de fósforos, bueno por cada dígito existe diferente cantidad de palitos requeridos (vea la imagen adjunta).



Dado una cadena representando el número que el niño desea construir, devolver el número de palitos que el niño requerirá para construir el número.

### Input

La primera línea del INPUT comienza con un número M ( $1 \leq M \leq 50$ ) que representa el número de casos, M líneas le siguen.

Cada siguiente línea contiene una cadena que contendrá entre 1 y 50 caracteres, inclusive.

Cada carácter será un dígito entre 0 y 9, inclusive.

### Output

Deberá formatear su salida de la forma "Caso #N: L".

(Comillas para claridad) donde N es el número de caso y L es un entero indicando la cantidad de palitos requeridos.

Input de ejemplo	Output correspondiente
3 13579 02468 73254370932875002027963295052175	Caso #1: 21 Caso #2: 28 Caso #3: 157

Caso 1: 1 consiste de 2 palitos; 3 consiste de 5 palitos; 5 consiste de 5 palitos; 7 consiste de 3 palitos; 9 consiste de 6 palitos;  
 $2 + 5 + 5 + 3 + 6 = 21$ .

# Problema B

## Sec2time

Los Computadores guardan fechas como un único número que representa el número de segundos o milisegundos desde una particular fecha. Tu tarea en este problema es tomar un entero como entrada que representa el número de segundos desde la media noche de algún día y devolver una cadena formateada así: "**H:M:S**". Donde **H** representa el número de horas completas desde la medianoche, **M** representa el número de minutos completados desde la última hora completa terminada, y **S** representa el número de segundos desde el último minuto completo terminado. Si tu recibes 0 segundos debes responder "0:0:0", mientras que si tu recibes 3661 tu deberías responder "1:1:1" (sin ceros adelante).

### Input

La primera línea del INPUT comienza con un número  $M$  ( $1 \leq M \leq 100$ ) que representa el número de casos,  $M$  líneas le siguen.

Cada siguiente línea contiene un entero  $N$  entre 0 y  $24 \cdot 60 \cdot 60 - 1 = 86399$ , inclusive.

### Output

Ud. Deberá formatear su salida de la siguiente forma

"Caso #N: S"

(Comillas para claridad) donde  $N$  es el número de caso y  $S$  es la cadena correctamente formateada.

Input de ejemplo	Output correspondiente
4	Caso #1: 0:0:0
0	Caso #2: 1:1:1
3661	Caso #3: 1:30:36
5436	Caso #4: 23:59:59
86399	

# Problema C

## NichoBits

Los computadores operan en números binarios. Casi toda la computación es hecha por la manipulación de 0s y 1s. De esta forma, para que los computadores puedan usar los números que nosotros les damos, ellos deben convertirlos de base 10 (lo que nosotros usamos normalmente) a binario (base 2). A veces es muy útil determinar cuantos bits usara un número para su representación, de tal manera que podamos ahorrar memoria. Por ejemplo, si un número es menor que 256, nosotros podemos representarlo con 8 bits.

Un valor de un número binario esta determinado de la siguiente manera: Para cada '1' en el número binario agrega  $2^i$  (2 a la i), donde i es el número de dígitos a la derecha del '1'. Por ejemplo, 10100 binario, es equivalente a 20 en decimal. El primer 1 tiene 4 dígitos a su derecha, entonces es equivalente a  $2^4 = 16$ . El otro 1 tiene dos dígitos a su derecha, entonces es  $2^2 = 4$ .  $16 + 4 = 20$ . Otro ejemplo es 1111, donde en base 10 es  $2^3 + 2^2 + 2^1 + 2^0 = 8 + 4 + 2 + 1 = 15$ .

Dado este conocimiento, Nicho, un profesor que le encanta los bits, quiere saber la cantidad mínima de bits que deben ser usados para representar un determinado número decimal, vamos, ayúdalo.

### Input

La primera línea del INPUT comienza con un número M ( $1 \leq M \leq 100$ ) que representa el número de casos, M líneas le siguen.

Cada siguiente línea contiene un entero entre 1 y 1000000, inclusive.

### Output

Ud. Deberá formatear su salida de la siguiente forma

"Caso #N: L"

(Comillas para claridad) donde N es el numero de caso y L la respuesta.

Input de ejemplo	Output
4 32 12 1 1500	Caso #1: 6 Caso #2: 4 Caso #3: 1 Caso #4: 11

Caso 1: 32 en binario es 100000 porque  $2^5 = 32$ , entonces un 1 mas cinco 0s a su derecha es 32. De esta manera necesitamos 6 dígitos para representarlo.

Caso 2: 12 en binario es 1100

# Problema D

## Puchos Divertidos

Una secuencia de caracteres son llamados *Puchos Divertidos* si consiste de `|` y `-` caracteres, como `|-|-|` o `-|-|` (comillas por claridad). Note que `|-||-|` o `---` no son *Puchos Divertidos*, porque contienen dos caracteres iguales adyacentes uno del otro. Dado una cadena, encuentre la más larga subcadena de toda la cadena que sea un *Pucho Divertido* y responda su longitud.

### Input

La primera línea del INPUT comienza con un número M (1<=M<=100) que representa el número de casos, M líneas le siguen.

Cada siguiente línea contiene una cadena que contiene entre 1 y 50 caracteres (inclusive). Cada carácter será solo `-` o `|`.

### Output

Ud. Deberá formatear su salida de la siguiente forma

"Caso #N: L"

(Comillas para claridad) donde N es el numero de caso y L es la longitud del mayor Pucho Divertido encontrado en la cadena de entrada.

Input de ejemplo	Output
6  - -  - - - -        - - -  - --- - --- -     - - -- -- --- - - - - - --- - - - -	Caso #1: 5 Caso #2: 7 Caso #3: 1 Caso #4: 4 Caso #5: 5 Caso #6: 8

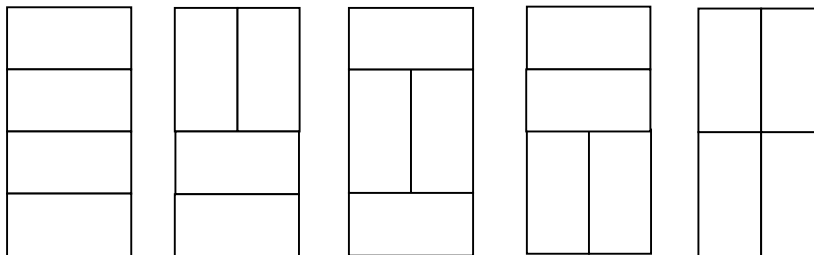
Caso 3: Un *Pucho Divertido* puede tener longitud 1, de esa forma un carácter ahí es un *Pucho Divertido*

# Problema E

## Dominó

Nosotros deseamos formar una red de 2 unidades de ancho y N unidades de largo, compuesto por rectángulos (dominós) de 2x1 unidades (en cualquier dirección).

Por ejemplo, en la figura, Para N=4 Hay 5 formas diferentes de hacer una red de 4x2 unidades.



La tarea consiste en determinar la mayor cantidad de formas diferentes para hacer una red de  $N \times 2$

### Input

La primera línea del INPUT comienza con un número M ( $1 \leq M \leq 100$ ) que representa el número de casos, M líneas le siguen.

Cada siguiente línea contiene el entero N ( $1 \leq N \leq 60$ ) al cual que se refiere el enunciado.

### Output

Deberá formatear su salida de la forma "Caso #X: L".

(Comillas para claridad) donde X es el numero de caso y L es un entero indicando la cantidad de formas diferentes de construir una red.

Input de ejemplo	Output correspondiente
4	Caso #1: 1
1	Caso #2: 2
2	Caso #3: 5
4	Caso #4: 6765
19	