

## Cadenas de divisores

Contribución de Matías Raimundez

### Descripción del problema

Agustín estaba pensando en el número 60, que es un número con muchos divisores. De repente se le ocurrió crear una cadena de divisores, en la que el primer elemento sea 60 y a partir de ahí cada elemento sea un divisor del anterior.

Por ejemplo, una posible cadena sería 60 - 20 - 10 - 2.

A Agustín le interesa que la suma de los elementos de la cadena sea lo más grande posible. Probó todas las cadenas de divisores que comienzan en 60, y descubrió que la cadena de divisores encabezada por 60 **con la mayor suma posible de sus elementos** es 60 - 30 - 15 - 5 - 1, cuya suma es 101.

Agustín comenzó con un número  $a = 60$  en el ejemplo anterior, pero la cadena podría comenzar con cualquier entero positivo  $a$ .

Debes escribir una función que calcule la máxima suma posible en una cadena de divisores, dado el número inicial  $a$  de la cadena.

### Descripción de la función

Debes implementar la función `cadenas(a)`

El único parámetro indica el número inicial  $a$  de la cadena. La función debe retornar la máxima suma posible de los elementos de una cadena que comience con el número  $a$ .

### Evaluador

El evaluador local lee de la entrada estándar un único entero, que indica el parámetro  $a$  para la función `cadenas`.

El evaluador local escribe a la salida estándar un único entero, con el valor retornado por la llamada `cadenas(a)`.

### Restricciones

- $1 \leq a \leq 1.000.000$

### Ejemplo

Si se invoca al evaluador con la siguiente entrada:

60

Para un programa correcto, la salida será:

101

Si en cambio la entrada es:

33

Para un programa correcto, la salida será:

45

### Subtareas

1.  $a \leq 10$  (10 puntos)
2.  $a \leq 5.000$  y  $a$  tiene a lo sumo 4 divisores (15 puntos)
3.  $a$  tiene a lo sumo 4 divisores (20 puntos)
4. Sin más restricción (55 puntos)