



Ejercicios entrenamiento para el concurso regional Programming.class 2013 que se celebrará en el IES Augustóbriga de Navalморal de la Mata el 18 de Abril del 2013

Con el apoyo de:



En este documento se especifican todos los detalles de cada uno de los ejercicios de la **SEGUNDA ENTREGA** que están subidos al juez online.

Los ejercicios no están colocados por orden de dificultad, es decir, puede ser que el último ejercicio de este documento sea más sencillo de solucionar que el primero.

Cada ejercicio tiene un nombre y, precediéndole y entre paréntesis, un identificador. Este identificador es el que utilizaremos en el juez online para encontrarlo.

Para acceder al juez utilizaremos la siguiente dirección:

<http://80.36.53.96/domjudge/team/>

Los usuarios y contraseñas para poder subir ejercicios al juez se proporcionarán por correo electrónico.



**TABLA DE EJERCICIOS**

<b>(DES_CUBO) DESCOMPOSICIÓN DE UN CUBO EN SUMA DE NÚMEROS IMPARES .....</b>	<b>5</b>
DESCRIPCIÓN DEL PROBLEMA.....	5
ENTRADA .....	5
SALIDA.....	5
ENTRADA DE EJEMPLO.....	6
SALIDA DE EJEMPLO .....	6
<b>(NUM_ELEG) CÁLCULO DE NÚMEROS ELEGANTES .....</b>	<b>7</b>
DESCRIPCIÓN DEL PROBLEMA.....	7
ENTRADA .....	7
SALIDA.....	7
ENTRADA DE EJEMPLO.....	8
SALIDA DE EJEMPLO .....	8
<b>(MDD) MATRICES DE DIAGONAL DOMINANTE.....</b>	<b>9</b>
DESCRIPCIÓN DEL PROBLEMA.....	9
ENTRADA .....	9
SALIDA.....	9
ENTRADA DE EJEMPLO.....	10
SALIDA DE EJEMPLO .....	10
<b>(CONCENTR) MATRIZ CONCÉNTRICA.....</b>	<b>11</b>
DESCRIPCIÓN DEL PROBLEMA.....	11
ENTRADA .....	11
SALIDA.....	11
ENTRADA DE EJEMPLO.....	12
SALIDA DE EJEMPLO .....	12
<b>(MCD) MÁXIMO COMÚN DIVISOR .....</b>	<b>13</b>
DESCRIPCIÓN DEL PROBLEMA.....	13
ENTRADA .....	13
SALIDA.....	13
ENTRADA DE EJEMPLO.....	13
SALIDA DE EJEMPLO .....	13
<b>(NUM_CURI) NUMEROS CURIOSOS .....</b>	<b>14</b>
DESCRIPCIÓN DEL PROBLEMA.....	14
ENTRADA .....	14
SALIDA.....	14
ENTRADA DE EJEMPLO.....	14
SALIDA DE EJEMPLO .....	14



<b>(NUM_NARC) NÚMERO NARCISTA.....</b>	<b>15</b>
DESCRIPCIÓN DEL PROBLEMA.....	15
ENTRADA .....	15
SALIDA.....	15
ENTRADA DE EJEMPLO.....	15
SALIDA DE EJEMPLO .....	15
<b>(NUM_PERF) NÚMERO PERFECTO .....</b>	<b>16</b>
DESCRIPCIÓN DEL PROBLEMA.....	16
ENTRADA .....	16
SALIDA.....	16
ENTRADA DE EJEMPLO.....	16
SALIDA DE EJEMPLO .....	16
<b>(NUM_TARO) NÚMERO DEL TAROT .....</b>	<b>17</b>
DESCRIPCIÓN DEL PROBLEMA.....	17
ENTRADA .....	17
SALIDA.....	17
ENTRADA DE EJEMPLO.....	17
SALIDA DE EJEMPLO .....	17
<b>(SEGUNDOS) TRANSFORMACIÓN EN SEGUNDOS .....</b>	<b>18</b>
DESCRIPCIÓN DEL PROBLEMA.....	18
ENTRADA .....	18
SALIDA.....	18
ENTRADA DE EJEMPLO.....	19
SALIDA DE EJEMPLO .....	19



# (DES\_CUBO) Descomposición de un cubo en suma de números impares

## Descripción del problema

Para descomponer el cubo de un número  $n$  en suma de números impares, se pueden sumar  $n$  números impares consecutivos a partir del número impar  $(n*(n-1)/2)+1$

Es posible descomponer un cubo en suma de números impares de manera muy sencilla: Para calcular:

$1^3 = 1$  (suma de **un** impar a partir del **primer** impar)

Empezando en  $\frac{1(1-1)}{2} + 1 = 1$

$2^3 = 3+5$  (suma de **dos** números impares a partir del **segundo** impar )

Empezando en  $\frac{2(2-1)}{2} + 1 = 2$

$3^3 = 7+9+11$  (suma de **tres** números impares a partir del **cuarto** impar )

Empezando en  $\frac{3(3-1)}{2} + 1 = 4$

$n^3 =$  (suma de **n** números impares desde el  $\frac{n(n-1)}{2} + 1$  **avo** número impar

Codificar un programa que reciba varios números como entrada y muestre su descomposición en números impares.

## Entrada

El primer número será el número de descomposiciones a realizar y a continuación se introducirán los números a descomponer.

## Salida

El programa deberá mostrar la salida con el siguiente formato:

$n^3 = v1 + v2 + v3 + \dots + vn = \text{valor\_}n^3$

Si el número introducido es menor que 1, el programa indicará "-x ES VALOR NO VALIDO" (sin acentos)

-x ES VALOR NO VALIDO



### Entrada de ejemplo

```
4
-1
1
3
7
```

### Salida de ejemplo

```
-1 ES VALOR NO VALIDO
1^3=1=1
3^3=7+9+11=27
7^3=43+45+47+49+51+53+55=343
```



# (NUM\_ELEG) Cálculo de números elegantes

## Descripción del problema

Para decidir si un número positivo es **elegante** se calcula la suma del cuadrado de sus cifras. Si utilizamos como ejemplo el número 9100, la suma de los cuadrados de sus cifras sería:

$$9^2 + 1^2 + 0^2 + 0^2 = 81 + 1 + 0 + 0 = 82$$

Repetimos la misma operación con los dígitos del nuevo número hasta que obtenemos uno de los siguientes resultados:

- 1: el número es elegante.
- 4: el número no es elegante.

Continuando con el ejemplo del número 9100:

- Iteración 1:  $9^2 + 1^2 + 0^2 + 0^2 = 81 + 1 + 0 + 0 = 82$
- Iteración 2:  $8^2 + 2^2 = 64 + 4 = 68$
- Iteración 3:  $6^2 + 8^2 = 36 + 64 = 100$
- Iteración 4:  $1^2 + 0^2 + 0^2 = 1 + 0 + 0 = 1$

El número 9100 es elegante

El número 16 no es elegante:

- Iteración 1:  $1^2 + 6^2 = 1 + 36 = 37$
- Iteración 2:  $3^2 + 7^2 = 9 + 49 = 58$
- Iteración 3:  $5^2 + 8^2 = 25 + 64 = 89$
- Iteración 4:  $8^2 + 9^2 = 64 + 81 = 145$
- Iteración 5:  $1^2 + 4^2 + 5^2 = 1 + 16 + 25 = 42$
- Iteración 6:  $4^2 + 2^2 = 16 + 4 = 20$
- Iteración 7:  $2^2 + 2^2 = 4 + 0 = 4$  El número no es elegante

## Entrada

La entrada está formada por una secuencia de números enteros positivos y mayores de 0. El primer valor indica cuantos números tiene la secuencia.

## Salida

Para cada uno de los números de entrada se crea una línea de texto de salida, que muestra SI si el número es elegante y NO si no lo es.



### Entrada de ejemplo

```
7
67
50
100
6
120
70
133
```

### Salida de ejemplo

```
NO
NO
SI
NO
NO
SI
SI
```



# (MDD) Matrices de diagonal dominante

## Descripción del problema

Para resolver sistemas de ecuaciones computacionalmente, hay ciertos métodos, que bajo ciertas condiciones, convergen muy rápido a la solución. Una de esas condiciones es cuando la matriz asociada a un sistema de ecuaciones es diagonalmente dominante.

Se dice que una matriz es diagonalmente dominante si por cada fila, el elemento de la diagonal principal es mayor que la suma del resto de elementos de su fila. Por ejemplo, la matriz M1 ES diagonalmente dominante por filas, y M2 NO LO ES:

M1

5	0	4
2	12	5
2	3	6

Elemento diagonal=5; Suma del resto:  $0 + 4 = 4$  ;  $5 > 4$ ?...OK

Elemento diagonal=12; Suma del resto:  $2 + 5 = 7$ ;  $12 > 7$ ?.. OK

Elemento diagonal=6; Suma del resto:  $3 + 2 = 5$ ;  $6 > 5$ ?... OK

**DIAGONALMENTE DOMINANTE!!!**

M2

5	0	4
4	12	8
2	3	6

Elemento diagonal=5; Suma del resto:  $0 + 4 = 4$ ; ¿ $5 > 4$ ?... OK

Elemento diagonal=12; Suma del resto:  $8 + 4 = 12$ ; ¿ $12 > 12$ ?.. NO!!!

Elemento diagonal=6; Suma del resto:  $3 + 2 = 5$

**MATRIZ NO DIAGONALMENTE DOMINANTE!!!**

Construir un programa que reciba matrices de cualquier dimensión e indicar si son diagonalmente dominantes. Se considerará como diagonal principal de la matriz aquellos elementos cuyo índice de fila sea igual al índice de columna.

## Entrada

El programa recibirá una primera línea con la dimensión de la matriz (filas x columnas) y en las siguientes líneas la matriz completa.

La entrada de datos terminará si la dimensión de la matriz es 0 x 0

## Salida

Si la matriz es diagonalmente dominante el proceso mostrará: ES MATRIZ DE DIAGONAL DOMINANTE

Si la matriz no es diagonalmente dominante el proceso terminará en la fila en la que se detecte que no cumple la condición de dominante mostrando el mensaje NO ES MATRIZ DE DIAGONAL DOMINANTE



### Entrada de ejemplo

```
3 3
5 0 4
2 12 5
2 3 6
3 3
5 0 4
4 12 8
2 3 6
0 0
```

### Salida de ejemplo

```
ES MATRIZ DE DIAGONAL DOMINANTE
NO ES MATRIZ DE DIAGONAL DOMINANTE
```



# (CONCENTR) Matriz Concéntrica

## Descripción del problema

Una matriz es concéntrica si, para todos sus anillos, los números correspondientes a cada anillo son iguales.

Ejemplo:

Matriz Concéntrica

3	3	3	3	3
3	7	7	7	3
3	7	5	7	3
3	7	7	7	3
3	3	3	3	3

Matriz NO Concéntrica

3	3	3	3	3
3	7	7	2	3
3	7	5	7	3
3	7	7	7	3
3	3	3	3	3

Realizar un algoritmo que reciba una matriz de 8 x 8 y muestre si se trata de una matriz concéntrica o no.

## Entrada

La entrada constará de matrices de 8 x 8 hasta que el primer dato de una matriz sea un cero.

## Salida

La salida será Si si la matriz es concéntrica y No si no lo es.





Entrada de ejemplo

```
1 2 3 4 5 6 7 8
2 3 4 5 6 7 8 9
12 3 4 5 6 77 8 9
3 4 5 6 7 8 9 0
1 2 3 4 5 6 7 8
2 3 4 5 6 7 8 9
3 4 5 6 7 8 9 0
1 2 3 4 5 6 7 8
3 3 3 3 3 3 3 3
3 5 5 5 5 5 5 3
3 5 7 7 7 7 5 3
3 5 7 9 9 7 5 3
3 5 7 9 9 7 5 3
3 5 7 7 7 7 5 3
3 5 5 5 5 5 5 3
3 3 3 3 3 3 3 3
30 30 30 30 30 30 30 30
30 50 50 50 50 50 50 30
30 50 70 70 70 70 50 30
30 50 70 90 90 70 50 30
30 50 70 90 90 70 50 30
30 50 70 70 70 70 50 30
30 50 50 50 50 50 50 30
30 30 30 30 30 30 30 30
3 3 3 3 3 3 3 3
3 5 2 5 5 5 5 3
3 5 7 7 7 7 5 3
3 5 7 9 9 7 5 3
3 5 7 9 9 7 5 3
3 5 7 7 7 7 5 3
3 5 5 5 5 5 5 3
3 3 3 3 3 3 3 3
0
```

Salida de ejemplo

```
No
Si
Si
No
```



# (MCD) Máximo Común Divisor

## Descripción del problema

El algoritmo de Euclides nos dice que el MCD de dos números A y B se calcula seleccionando el menor de los dos números y depositándolo en A y seleccionando el resto de la división de A y B y depositándolo en B, realizando este proceso hasta que B tome el valor 0 en cuyo caso el MCD será A. Este proceso es independiente de que A sea mayor que B, B mayor que A o A y B iguales.

Ejemplo:

A	B	MENOR	RESTO
728	304	304	120
304	120	120	64
120	64	64	56
64	56	56	8
56	8	8	0
8	0	MCD = 8	

Realizar un algoritmo que reciba dos números enteros positivos no nulos y devuelva su MCD. La entrada serán parejas de números de las que queremos calcular el MCD y finalizará cuando uno de los dos números sea cero.

## Entrada

El algoritmo recibirá dos números enteros positivos no nulos hasta que uno de los dos números sea cero.

## Salida

El algoritmo devolverá el MCD de cada pareja de datos de entrada.

## Entrada de ejemplo

```
728 304
25 125
50 125
8 16
27 81
0 0
```

## Salida de ejemplo

```
8
25
25
8
27
```



## (NUM\_CURI) Numeros curiosos

### Descripción del problema

Llamaremos números curiosos a los que cumplen la siguiente propiedad: la suma del factorial de sus dígitos es igual al número.

Un ejemplo de número curioso es el 145. Si calculamos la suma del factorial de sus dígitos obtenemos:  $1! + 4! + 5! = 1 + 24 + 120 = 145$

### Entrada

La entrada consiste en una serie de números naturales que finalizará con la inserción del número 0.

### Salida

Formato

Especifica si se trata o no de un número curioso.

Se visualizará por pantalla un mensaje indicando la afirmación o negación del número curioso.

El formato del mensaje será:

número ES UN NUMERO CURIOSO

número NO ES UN NUMERO CURIOSO

### Entrada de ejemplo

```
2934
77
145
23
2
34340
0
```

### Salida de ejemplo

```
2934 NO ES UN NUMERO CURIOSO
77 NO ES UN NUMERO CURIOSO
145 ES UN NUMERO CURIOSO
23 NO ES UN NUMERO CURIOSO
2 ES UN NUMERO CURIOSO
34340 NO ES UN NUMERO CURIOSO
```



## (NUM\_NARC) Número narcisista

### Descripción del problema

Se conoce como número narcisista aquel número de  $n$  dígitos que resulta ser igual a la suma de las potencias de orden  $n$  de los dígitos que lo componen.

Ejemplo número 153, es narcisista ya que:

Potencia orden:  $n=3$

$$153 = 1^3 + 5^3 + 3^3 = 1 + 125 + 27.$$

Codificar un programa que reciba varios números como entrada y muestre si son narcisistas.

### Entrada

El programa recibirá una sucesión de número naturales, el programa acabará cuando se introduzca un 0.

### Salida

El programa deberá mostrar "Si" si el número es narcisista o "No" en caso contrario

### Entrada de ejemplo

```
25
153
12
6
0
```

### Salida de ejemplo

```
No
Si
No
Si
```



## (NUM\_PERF) Número perfecto

### Descripción del problema

Un número  $X$  es perfecto si la suma de sus divisores propios, sin incluirse  $X$ , da como resultado el mismo número  $X$ . Teniendo en cuenta que los divisores propios son aquellos números que dividiendo a  $X$  dan de resto cero.

Por ejemplo:

El 6 es perfecto pues  $6 = 2 + 3 + 1$  (4, 3 y 1 son los divisores propios pues  $6/2$  da resto 0, igual que  $6/3$  y  $6/1$ )

El 8 no es perfecto pues 8 es distinto de  $4 + 2 + 1$

Codificar un programa que reciba varios números como entrada y muestre si son perfectos.

### Entrada

El programa recibirá una sucesión de número naturales, el programa acabará cuando aparezca un 0

### Salida

El programa deberá mostrar "Si" si el número es perfecto o "No" en caso contrario

### Entrada de ejemplo

```
12
6
29
0
```

### Salida de ejemplo

```
No
Si
No
```





## (NUM\_TARO) Número del tarot

### Descripción del problema

La numerología estudia el significado e influencia de los números en la vida de los seres humanos. Es una de las ciencias ocultas que la humanidad ha cultivado desde el más lejano pasado.

De todos los números que integran nuestra vida, es especialmente importante el de nuestra fecha de nacimiento. El *Número del Tarot* se calcula sumando los dígitos de la fecha de nacimiento y reduciéndolos sucesivamente a un único dígito (del 1 al 9).

Por ejemplo, para la fecha 17 de octubre de 1970, los cálculos serían:

$$1+7+1+0+1+9+7+0=26$$

$$2+6=8$$

### Entrada

El programa recibirá una cantidad determinada de *fechas de nacimiento*, todas válidas, con el formato *dd mm aaaa*. El programa termina cuando la entrada sea -1.

### Salida

El programa devolverá el número del tarot para cada fecha de la entrada.

### Entrada de ejemplo

```
17 10 1970
28 02 2006
01 01 2001
29 09 1989
18 07 2009
-1
```

### Salida de ejemplo

```
8
2
5
2
9
```



# (SEGUNDOS) Transformación en segundos

## Descripción del problema

Realizar un programa que transforme un número determinado de segundos en años, meses, días, horas, minutos y segundos. Para ello hay que tener en cuenta los siguientes datos:

- 1 minuto = 60 segundos
- 1 hora = 3600 segundos
- 1 día = 86400 segundos
- 1 mes (de 30 días) = 2592000 segundos
- 1 año (de 12 meses)= 31104000 segundos

Consideraremos que los meses siempre tienen 30 días y los años 12 meses, de 30 días cada uno.

## Entrada

Como entrada se pasará al programa el número de segundos que desea transformar, este valor estará comprendido entre 1 y 2147483647. El proceso se repetirá hasta que se introduzcan 0 segundos.

## Salida

La salida tendrá el formato `unidad_de_tiempo=cantidad_de_segundos` y, cada unidad de tiempo aparecerá en una línea distinta. No existirán espacios antes ni después del símbolo `=`. Cada unidad de tiempo se indicará con su primera letra en minúscula, salvo en los meses que será en mayúscula para diferenciarlos de los minutos: `a` para años, `M` para meses, `d` para días, `h` para horas, `m` para minutos y `s` para segundos. Las unidades que tengan el valor 0 no se mostrarán.



### Entrada de ejemplo

```
60
4000
150002300
0
```

### Salida de ejemplo

```
60
m=1
4000
h=1
m=6
s=40
150002300
a=4
M=9
d=26
h=3
m=18
s=20
```

