

# { code } { FEST }



SEVILLA, 11 DE NOVIEMBRE DE 2016

## Reto 1

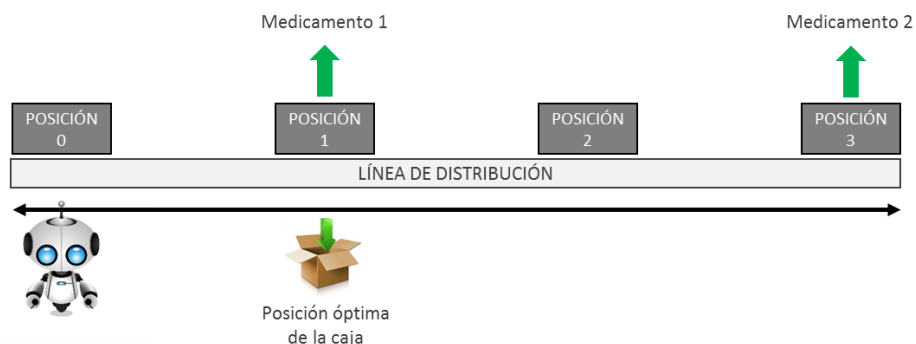
### Línea robótica de distribución farmacéutica

TIEMPO DISPONIBLE: 60 MINUTOS

En una cooperativa farmacéutica estamos instalando una línea automatizada de distribución de medicamentos. En su funcionamiento, un robot va tomando medicamentos de distintas posiciones de la línea y los va almacenando en una caja para su almacenamiento y distribución posterior por el servicio de transporte.

Para ello, necesitamos diseñar un algoritmo que permita decidir la posición óptima de la caja, que minimice los movimientos del robot en la línea, sabiendo previamente la ubicación de los medicamentos a distribuir.

El robot se desplaza siempre en línea recta, tanto para ubicar la caja al comienzo, como para recoger cada medicamento en su posición e incluirlo en la caja de uno en uno. Las posiciones consecutivas de la línea están separadas por una distancia de un metro. El robot empieza y termina siempre en la posición cero, que es donde se encuentran las cajas al comienzo. Si hay varios medicamentos en una misma posición será necesario ubicarlos de uno en uno. La posición óptima de ubicación de la caja debe ser un número entero, es decir, no es posible colocar la caja en posiciones intermedias.



Por ejemplo, para 2 medicamentos en las posiciones {1 y 3} (uno de ellos en la posición 1 y otro en la posición 3), la posición óptima de la caja es la 1, y el robot recorrerá la distancia de 6 metros:

- 1 metro para colocar la caja en la posición 1 partiendo desde la posición 0.
- 0 metros para ubicar en la caja el medicamento de la posición 1.
- 2 metros + 2 metros para ubicar en la caja el medicamento de la posición 3.
- 1 metro para volver a la posición 0.

El reto consiste en desarrollar un algoritmo que minimice y calcule el total de metros recorridos para distribuir un conjunto de medicamentos cuyas posiciones son conocidas.

Algunos ejemplos para que puedas probar tu algoritmo:

Nº de medicamentos	Posiciones	Distancia óptima recorrida (en metros)
2	{1, 3}	6
6	{2, 3, 5, 8, 9, 9}	42
12	{1, 1, 1, 3, 4, 9, 15, 19, 19, 21, 22, 25}	222

Criterios de valoración:

- El ganador del reto será la primera persona que implemente el algoritmo indicado, informando del resultado que obtiene para el siguiente caso:

Nº de medicamentos	Posiciones	Distancia óptima recorrida (en metros)
20	{1, 3, 4, 8, 8, 10, 10, 14, 15, 16, 20, 21, 22, 22, 23, 24, 24, 27, 31, 39}	¿?

- A la finalización del codeFEST se nombrará a la "escuela revelación 2016", como la escuela que haya conseguido más puntos entre sus alumnos. El ganador de este reto obtendrá 10 puntos, que acumularán por su escuela para obtener esta mención.

Forma de entrega:

- Antes de entregar asegúrate de que tu algoritmo obtiene resultados correctos con los datos de ejemplo que se han indicado antes.
- Si es correcto comunica tu resultado (distancia óptima recorrida en metros) en Twitter con el siguiente formato:  
**@everisCodeFEST #mireto1 <escribe aquí la distancia óptima recorrida>**
- Si por algún motivo no tuvieras conectividad a Twitter, ponte en contacto con tu dinamizador asignado, que te ayudará publicar el resultado.
- El ganador se decidirá según orden de entrada en Twitter con el hashtag indicado.

¡¡Suerte!!

# { code } { FEST }



SEVILLA, 11 DE NOVIEMBRE DE 2016

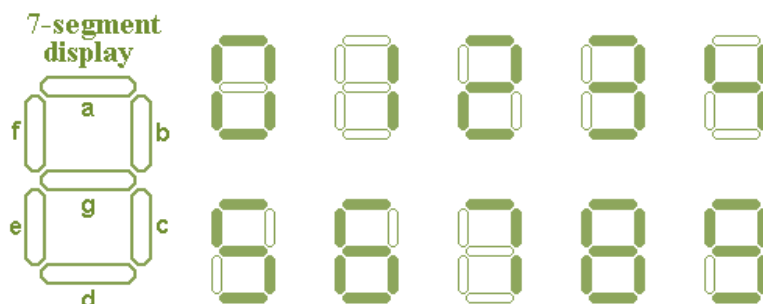
## Reto 2

### Consumo de un dispositivo electrónico

TIEMPO DISPONIBLE: 60 MINUTOS

En una compañía de diseño de dispositivos electrónicos, se necesita desarrollar una herramienta para simular el consumo eléctrico de un display que posteriormente se integrará como parte de un reloj digital.

El display tiene 6 dígitos, para representar horas, minutos y segundos, contando desde 00:00:00 hasta 23:59:59. Y cada dígito se implementará con un dispositivo de 7 segmentos que, mediante diodos led, codifica los números de la siguiente manera:



Al tratarse de un reloj, los dígitos se mantienen encendidos y cambian como lo hace un reloj, cada segundo. La energía consumida por el diodo led de cada segmento cuanto está encendido durante un segundo es de 1 (Wh).

Así, por ejemplo:

- El dígito '1', al tener 2 segmentos encendidos, consume 2 Wh en un segundo.
- El dígito '2', al tener 5 segmentos encendidos, consume 5 Wh cada segundo.

El reto consiste en desarrollar un algoritmo que calcule el consumo en Wh del display de 6 dígitos cuando está funcionando correctamente durante un año completo de 365 días.

Algunos ejemplos para que puedas probar tu algoritmo:

Dígitos del display	Tiempo funcionando	Energía consumida (Wh)
00:00:00 – 00:00:09	10 segundos	349
00:00:00 – 00:00:39	40 segundos	1336
00:00:00 – 00:14:19	14 minutos y 20 segundos	26614

Criterios de valoración:

- El ganador del reto será la primera persona que implemente el algoritmo indicado, informando del resultado que obtiene para el siguiente caso:

Dígitos del display	Tiempo funcionando	Energía consumida (Wh)
00:00:00 – 23:59:59 cada día	365 días	¿?

- A la finalización del codeFEST se nombrará a la “escuela revelación 2016”, como la escuela que haya conseguido más puntos entre sus alumnos. El ganador de este reto obtendrá 11 puntos, que acumularán por su escuela para obtener esta mención.

Forma de entrega:

- Antes de entregar asegúrate de que tu algoritmo obtiene resultados correctos con los datos de ejemplo que se han indicado antes.
- Si es correcto comunica tu resultado (energía consumida en Wh) en Twitter con el siguiente formato:  
**@everisCodeFEST #mireto2 <escribe aquí la energía consumida>**
- Si por algún motivo no tuvieras conectividad a Twitter, ponte en contacto con tu dinamizador asignado, que te ayudará publicar el resultado.
- El ganador se decidirá según orden de entrada en Twitter con el hashtag indicado.

¡¡Suerte!!

# { code } { FEST }



SEVILLA, 11 DE NOVIEMBRE DE 2016

## Reto 3

### Cifrado por descomposición de factores

TIEMPO DISPONIBLE: 60 MINUTOS

Para desarrollar un sistema de cifrado, se necesita implementar un algoritmo que identifique los números enteros que se puedan obtener combinando 2 tipos de operaciones básicas a partir del número 1:

- Sumar 5.
- Multiplicar por 3.

A continuación se muestran algunos ejemplos:

- $3 = 1 \times 3$
- $6 = 1 + 5$
- $8 = (1 \times 3) + 5$
- $24 = (1 + 5) \times 3$
- $9 = (1 \times 3) \times 3$
- $11 = (1 + 5) + 5$
- $14 = ((1 \times 3) \times 3) + 5$

Mientras que hay otros números enteros, que no se pueden obtener por descomposición de estas operaciones, como son el 2, el 4 o el 5 por ejemplo.

El reto consiste en desarrollar un algoritmo que calcule el número de enteros, mayores que uno y menores o iguales a un máximo dado, que se puedan descomponer en las operaciones indicadas.

Algunos ejemplos para que puedas probar tu algoritmo:

Máximo	Número de enteros menores o iguales, que se pueden descomponer
2	0
3	1
8	3
100	73
500	393

Criterios de valoración:

- El ganador del reto será la primera persona que implemente el algoritmo indicado, informando del resultado que obtiene para el siguiente caso:

Máximo	Número de enteros menores o iguales, que se pueden descomponer
7.000	¿?

- A la finalización del codeFEST se nombrará a la “escuela revelación 2016”, como la escuela que haya conseguido más puntos entre sus alumnos. El ganador de este reto obtendrá 12 puntos, que acumularán por su escuela para obtener esta mención.

Forma de entrega:

- Antes de entregar asegúrate de que tu algoritmo obtiene resultados correctos con los datos de ejemplo que se han indicado antes.
- Si es correcto comunica tu resultado (número de enteros menores o iguales, que se pueden descomponer en las operaciones indicadas) en Twitter con el siguiente formato:  
**@everisCodeFEST #mireto3 <escribe aquí tu resultado>**
- Si por algún motivo no tuvieras conectividad a Twitter, ponte en contacto con tu dinamizador asignado, que te ayudará publicar el resultado.
- El ganador se decidirá según orden de entrada en Twitter con el hashtag indicado.

¡¡Suerte!!

# { code } { FEST }



SEVILLA, 11 DE NOVIEMBRE DE 2016

## Reto 4

### Gestión de contenedores en entidad portuaria

TIEMPO DISPONIBLE: 60 MINUTOS

En una entidad portuaria, se requiere la implementación de un software para optimizar la ubicación de pedidos en contenedores para su transporte.

El objetivo es componer cada contenedor con pedidos que tengan el máximo precio de venta en el stock acumulado, sin sobrepasar el peso máximo autorizado para un contenedor, que es de 8.000 kilos.

El espacio ocupado no es un problema en este caso, ya que se dispone de contenedores de distintas alturas para adaptarse a la composición calculada.

Por ejemplo, para los siguientes pedidos:

Pedido	Peso (Kg)	Precio (euros)
Pedido 1	7.000	1.500.000
Pedido 2	1.000	70.000
Pedido 3	400	400.000
Pedido 4	650	300.000
Pedido 5	500	200.000

El contenedor óptimo tiene los siguientes pedidos {1, 3 y 5}, con un precio total de 2.100.000 euros en precios y 7.900 Kg de peso.

El reto consiste en desarrollar un algoritmo que decida los pedidos a ubicar en el contenedor para maximizar la suma de precios sin sobrepasar el peso máximo de 8.000 Kg.

A continuación se expone otro ejemplo más para que puedas probar tu algoritmo:

Pedido	Peso (Kg)	Precio (euros)
Pedido 1	5.000	900.000
Pedido 2	100	750.000
Pedido 3	3000	300.000
Pedido 4	300	50.000
Pedido 5	1500	500.000
Pedido 6	500	400.000
Pedido 7	900	10.000
Pedido 8	800	8.000

En este caso el contenedor óptimo tiene los siguientes pedidos {1, 2, 4, 5 y 6}, con un precio total de 2.600.000 euros en precios y 7.400 Kg de peso.

Criterios de valoración:

- El ganador del reto será la primera persona que implemente el algoritmo indicado, informando del resultado que obtiene para el siguiente caso:  
Tabla de datos en el fichero: <http://bit.ly/2fb465g>
- A la finalización del codeFEST se nombrará a la “escuela revelación 2016”, como la escuela que haya conseguido más puntos entre sus alumnos. El ganador de este reto obtendrá 13 puntos, que acumularán por su escuela para obtener esta mención.

Forma de entrega:

- Antes de entregar asegúrate de que tu algoritmo obtiene resultados correctos con los datos de ejemplo que se han indicado antes.
- Si es correcto comunica tu resultado (precio total del contenedor óptimo en euros) en Twitter con el siguiente formato:  
**@everisCodeFEST #mireto4 <escribe aquí el precio total óptimo>**
- Si por algún motivo no tuvieras conectividad a Twitter, ponte en contacto con tu dinamizador asignado, que te ayudará publicar el resultado.
- El ganador se decidirá según orden de entrada en Twitter con el hashtag indicado.

¡¡Suerte!!



# { code } { FEST }



SEVILLA, 11 DE NOVIEMBRE DE 2016

## Reto 5

### Planificación logística de almacenes

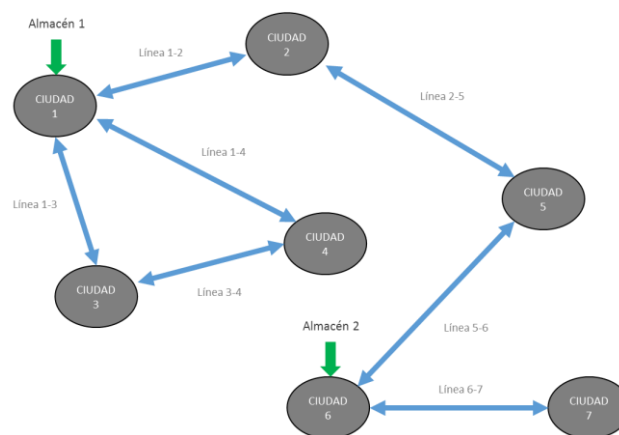
TIEMPO DISPONIBLE: 60 MINUTOS

Para un servicio de transporte, necesitamos desarrollar un sistema de planificación almacenes para dar servicio a distintos comercios de un área geográfica.

Para ello, tendremos identificadas las ciudades con comercios a los que debemos dar servicio, y sabremos las conexiones de transporte que hay entre ellas, que serán bidireccionales.

El sistema deberá calcular el número mínimo de almacenes a instalar, proponiendo en qué ciudades, sabiendo que cada almacén puede dar servicio a los comercios de su ciudad pero también a los de ciudades con las que exista una conexión de transporte directo.

A continuación se muestra un ejemplo con 7 ciudades y 7 conexiones de transporte:



En el ejemplo puede verse que con 2 almacenes puede darse servicio a todas las ciudades:

- El almacén 1 está ubicado en la ciudad 1 y da servicio a las ciudades {1, 2, 3 y 4}.
- El almacén 2 está ubicado en la ciudad 6 y da servicio a las ciudades {5, 6 y 7}.

El reto consiste en desarrollar un algoritmo que calcule el mínimo número de almacenes, indicando las ciudades propuestas.

A continuación se expone otro ejemplo más para que puedas probar tu algoritmo:

Conexión de transporte (todas bidireccionales)	Ciudad origen	Ciudad destino
Línea 1-2	1	2
Línea 2-3	2	3
Línea 3-4	3	4
Línea 2-5	2	5
Línea 5-6	5	6
Línea 6-7	6	7
Línea 1-8	1	8
Línea 8-9	8	9
Línea 6-10	6	10
Línea 10-11	10	11
Línea 10-12	10	12

En este caso el número óptimo de almacenes es 4 y la ubicación propuesta ellos es en las ciudades {3, 6, 8 y 10}.

Criterios de valoración:

- El ganador del reto será la primera persona que implemente el algoritmo indicado, informando del resultado que obtiene para el siguiente caso:  
Tabla de datos en el fichero: <http://bit.ly/2fngGBk>
- A la finalización del codeFEST se nombrará a la “escuela revelación 2016”, como la escuela que haya conseguido más puntos entre sus alumnos. El ganador de este reto obtendrá 14 puntos, que acumularán por su escuela para obtener esta mención.

Forma de entrega:

- Antes de entregar asegúrate de que tu algoritmo obtiene resultados correctos con los datos de ejemplo que se han indicado antes.
- Si es correcto comunica tu resultado (ciudades con almacén, por ejemplo: 3, 6, 8 y 10) en Twitter con el siguiente formato:  
**@everisCodeFEST #mireto5 <escribe aquí las ciudades con almacén>**
- Si por algún motivo no tuvieras conectividad a Twitter, ponte en contacto con tu dinamizador asignado, que te ayudará publicar el resultado.
- El ganador se decidirá según orden de entrada en Twitter con el hashtag indicado.

¡¡Suerte!!