

{ code } { FEST }



SEVILLA, 3 DE NOVIEMBRE DE 2017

Reto 1

Sistema de seguridad en comunicaciones

TIEMPO DISPONIBLE: 60 MINUTOS

Una compañía de telecomunicaciones está implementando un nuevo algoritmo de cifrado, que funciona de la siguiente manera:

- Parte de un array de datos formado por números enteros. La primera posición la ocupa el número 2, la segunda posición el número 3, la tercera posición el número 4 y así sucesivamente.
- Comienza en una posición concreta del array (índice p).
- Si el valor del elemento en esa posición es un número primo, se avanza a la posición del siguiente elemento ($p+1$).
- Si no es un número primo:
 - Obtiene el máximo divisor entero (q).
 - Divide el valor del elemento actual entre q .
 - Retrocede a la posición del elemento anterior ($p-1$).
 - Y suma q al valor de este elemento.
- Estas operaciones se repiten sucesivamente n veces.

NOTA 1: Un número entero es "primo" cuando solamente es divisible por sí mismo y por uno.

NOTA 2: El "máximo divisor" de un número no primo es el máximo de los enteros por los que es divisible.

A continuación se muestra un ejemplo de aplicación del algoritmo:

- Para el siguiente array: $A = [2, 3, 4, 5, 6, 7, \dots]$.
- Si inicialmente $p = 0$, $A[0] = 2$.
- Para $n = 1 \rightarrow$ como 2 es primo, se avanza a $p = p + 1 = 1$, $A[1] = 3$.
- Para $n = 2 \rightarrow$ como 3 es primo, se avanza a $p = p + 1 = 2$, $A[2] = 4$.
- Para $n = 3 \rightarrow$ como 4 no es primo, su máximo divisor es $q = 2$ y:
 - Se actualiza $A[2]$ por $A[2] / 2 = 2$.
 - Se actualiza $A[1]$ por $A[1] + 2 = 5$.

- Queda $A = [2, 5, 2, 5, 6, 7, \dots]$, $p = 1$, $A[1] = 5$.
- Para $n = 4 \rightarrow$ como 5 es primo, se avanza a $p = p + 1 = 2$, $A[2] = 2$.
- Para $n = 5 \rightarrow$ como 2 es primo, se avanza a $p = p + 1 = 3$, $A[3] = 5$.
- Para $n = 6 \rightarrow$ como 5 es primo, se avanza a $p = p + 1 = 4$, $A[4] = 6$.
- Para $n = 7 \rightarrow$ como 6 no es primo, su máximo divisor es $q = 3$ y:
 - Se actualiza $A[4]$ por $A[4] / 3 = 2$.
 - Se actualiza $A[3]$ por $A[3] + 3 = 8$.
 - Queda $A = [2, 5, 2, 8, 2, 7, \dots]$ y $p = 3$.

El reto consiste en desarrollar un algoritmo que, para el array indicado en el ejemplo, calcule la posición resultante (p), después de un número de iteraciones (n), cuando el algoritmo comienza en una posición inicial determinada (p_0).

Algunos ejemplos para que puedas probar tu algoritmo:

Posición inicial (p_0)	Nº de iteraciones (n)	Posición resultante (p)
0	5	3
0	7	3
9	5	12
9	10	11

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:

Posición inicial (p_0)	Nº de iteraciones (n)	Posición resultante (p)
21	1.000.000	¿?

- A la finalización del codeFEST se nombrará a la "escuela revelación 2017", 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 en Twitter con el siguiente formato:

@everisCodeFEST #mireto1 <posición resultante p> (indica tu nº de mesa)

Ejemplo: @everisCodeFEST #mireto1 3 (mesa 15)

- Si por algún motivo no tuvieras conectividad con Twitter, ponte en contacto con tu dinamizador asignado, que te ayudará a publicar el resultado.
- El ganador se decidirá según orden de entrada en Twitter con el hashtag indicado.

¡¡Suerte!!

{ code } { FEST }



SEVILLA, 3 DE NOVIEMBRE DE 2017

Reto 2

Alertas de búsqueda sobre documentación no estructurada

TIEMPO DISPONIBLE: 60 MINUTOS

Una compañía de seguros está implementando un sistema que envía alertas por email a sus clientes, cuando estos aparecen referenciados en el Boletín Oficial del Estado (BOE).

Para ello, necesitamos desarrollar un componente software que, a partir del texto extraído de cada documento PDF, identifique los números de Documento Nacional de Identidad (DNI) que sean correctos, seleccione sólo aquellos que están en la base de datos de clientes y prepare un fichero de texto con la información de los correos electrónicos a enviar.

Para que un número de DNI sea correcto debe estar compuesto por 8 dígitos decimales y un código de control. El código de control se obtiene a partir del número completo del DNI (8 cifras), dividiéndolo entre 23. Al resto resultante de dicha división, que deberá estar comprendido entre 0 y 22, se le asigna la letra de control según la equivalencia siguiente:

resto de la división ->	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22
código de control ->	T	R	W	A	G	M	Y	F	P	D	X	B	N	J	Z	S	Q	V	H	L	C	K	E

La base de datos de clientes tiene una tabla con los campos: nº de DNI, nombre, apellido 1, apellido 2 y dirección de email.

El texto a generar para cada email debe incluirse en un fichero de texto, con el siguiente formato para cada cliente:

```
To: <dirección de email>
Body: Hola <nombre> <apellido 1> <apellido 2>: nos ponemos contacto para indicarte que nuestro sistema de alarmas ha detectado que tu DNI nº <nº de DNI> aparece en el BOE de la fecha de hoy. Un saludo.
```

Ejemplo:

- Si en el texto de entrada encontramos los siguientes códigos: 10000949C, 10000320N, 20000717D, 1823Y, 20000128Z → Sólo seleccionaremos los 3 primeros, ya que el cuarto no tiene 8 dígitos decimales y el quinto tiene el código de control incorrecto.
- Si nuestra base de datos de clientes tiene los siguientes registros, tomaremos sólo los 2 primeros, que son los encontrados en el BOE:

DNI	Nombre	Apellido 1	Apellido 2	Email
10000949C	Olga	Miguel	Chao	olga@email.com
10000320N	María Encina	Feliz	Varela	mariaencina@email.com
30000831E	Pedro	García	Giraldo	pedro@email.com

- El fichero generado tendrá el siguiente texto:

```
To: olga@email.com
Body: Hola Olga Miguel Chao: nos ponemos contacto para indicarte que nuestro sistema de alarmas ha detectado que tu DNI nº 10000949C aparece en el BOE de la fecha de hoy. Un saludo.

To: mariaencina@email.com
Body: Hola María Encina Feliz Varela nos ponemos contacto para indicarte que nuestro sistema de alarmas ha detectado que tu DNI nº 10000320N aparece en el BOE de la fecha de hoy. Un saludo.
```

El reto consiste en desarrollar un algoritmo que genere el fichero de correos electrónicos a enviar y obtenga el número de clientes distintos a los que se va a enviar email, para el siguiente texto de entrada del BOE (<https://goo.gl/vvpMEP>).

Algunos ejemplos para que puedas probar tu algoritmo sobre la base de datos de clientes proporcionada en el siguiente fichero: <https://goo.gl/oyJUjp>.

Base de datos de clientes	Nº de clientes distintos
Caso 1 del fichero	1
Caso 2 del fichero	4

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, **habiendo generado correctamente el fichero de emails correspondiente:**

Base de datos de clientes	Nº de clientes distintos
Caso del reto a resolver en el fichero	¿?

- A la finalización del codeFEST se nombrará a la “escuela revelación 2017”, 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 (nº de clientes distintos a los que se envía email) en Twitter con el siguiente formato:

@everisCodeFEST #mireto2 <nº de clientes distintos> (indica tu nº de mesa)

Ejemplo: @everisCodeFEST #mireto2 3 (mesa 15)

- Si por algún motivo no tuvieras conectividad con Twitter, ponte en contacto con tu dinamizador asignado, que te ayudará a publicar el resultado.
- El ganador se decidirá según orden de entrada en Twitter con el hashtag indicado.

¡¡Suerte!!

{ code } { FEST }



SEVILLA, 3 DE NOVIEMBRE DE 2017

Reto 3

Automatización de almacén

TIEMPO DISPONIBLE: 60 MINUTOS

Una compañía de logística está desarrollando un sistema de planificación para su flota de vehículos en almacén.

Para ello, mantiene actualizada una previsión diaria con los horarios de llegada y salida de cada vehículo. Y en una de las funcionalidades a implementar se requiere conocer cuál es el período de tiempo en que el número de vehículos es el máximo en almacén.

El reto consiste en diseñar un algoritmo que, a partir de las horas previstas de llegada y salida de vehículos, calcule las horas de inicio y fin del período de máxima ocupación.

A continuación se muestra un ejemplo:

Vehículo	1	2	3	4	5	6	7	8
Llegada	10:00	12:00	11:00	13:00	14:00	12:00	9:00	14:00
Salida	15:00	14:00	17:00	15:00	15:00	16:00	14:00	15:00

En este caso, se puede comprobar que el número máximo de vehículos es 6 y se alcanza en el período de 13:00 a 15:00.

	1	2	3	4	5	6	7	8	Nº
9:00							█		1
10:00	█						█		2
11:00	█		█				█		3
12:00	█	█	█			█	█		5
13:00	█	█	█	█		█	█		6
14:00	█		█	█	█	█		█	6
15:00			█			█			2
16:00			█						1

Algunos ejemplos para que puedas probar tu algoritmo sobre los horarios proporcionados en el siguiente fichero: <https://goo.gl/zcVX6w>.

Caso	Nº máximo de vehículos	Período de máxima ocupación
Caso 1	8	14:22 - 14:27
Caso 2	5	13:49 - 14:04

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:

Caso	Nº máximo de vehículos	Período de máxima ocupación
Caso del reto a resolver	¿?	¿?

- A la finalización del codeFEST se nombrará a la “escuela revelación 2017”, 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 (período de máxima ocupación) en Twitter con el siguiente formato:

@everisCodeFEST #mireto3 <período> (indica tu nº de mesa)

Ejemplo: @everisCodeFEST #mireto3 13:00–15:00 (mesa 15)

- Si por algún motivo no tuvieras conectividad con Twitter, ponte en contacto con tu dinamizador asignado, que te ayudará a publicar el resultado.
- El ganador se decidirá según orden de entrada en Twitter con el hashtag indicado.

¡¡Suerte!!

{ code } { FEST }



SEVILLA, 3 DE NOVIEMBRE DE 2017

Reto 4

Optimización de rutas aéreas

TIEMPO DISPONIBLE: 60 MINUTOS

Una compañía de transporte aéreo está diseñando un nuevo software para determinar la ruta óptima de salida en situaciones de emergencia.

Cuando se detecta una situación de emergencia, el software debe construir un mapa a partir de la información del radar, donde aparecen los fenómenos meteorológicos y otros obstáculos a evitar. Este mapa se define en forma de matriz, con m filas, n columnas y $m * n$ cuadrículas.

El avión, en cada paso en su ruta, puede moverse de una celda en los siguientes sentidos: izquierda, derecha, arriba y abajo. No puede hacer movimientos en diagonal o a celdas no consecutivas. Tampoco se puede mover por las cuadrículas a evitar, que están marcadas sobre dicha matriz.

El reto consiste en diseñar un algoritmo que, a partir de la información matricial proporcionada por el radar con las cuadrículas a evitar, calcule la ruta más corta posible que lleve al avión fuera de la zona de peligro. El avión siempre sale de la cuadrícula central del mapa, y se considera que ha salido de peligro cuando alcanza cualquiera de los laterales de la matriz.

Por ejemplo, en el siguiente mapa las cuadrículas a evitar están marcadas en gris y la ruta más corta está coloreada en amarillo, con una longitud de 6 cuadrículas, partiendo desde la posición central y saliendo por la file 6 y columna 5. No existe una ruta más corta.

	0	1	2	3	4	5	6
0	gray	green	gray	green	gray	green	green
1	green	gray	green	green	gray	gray	green
2	green	green	green	gray	green	gray	green
3	gray	green	gray	*	gray	gray	green
4	gray	green	green	yellow	yellow	gray	gray
5	green	gray	green	gray	yellow	yellow	gray
6	green	green	gray	green	gray	yellow	green

Algunos ejemplos para que puedas probar tu algoritmo sobre el mapa proporcionado en el siguiente fichero: <https://goo.gl/t5JK2Y>. En la matriz de datos están marcados con "1" las cuadrículas a evitar y con "0" las restantes.

Mapa	Resultado: Longitud de la ruta más corta (número de cuadrículas)
Caso 1	6
Caso 2	7

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:

Mapa	Resultado: Longitud de la ruta más corta (número de cuadrículas)
Caso del reto a resolver	¿?

- A la finalización del codeFEST se nombrará a la "escuela revelación 2017", como la escuela que haya conseguido más puntos entre sus alumnos. El ganador de este reto obtendrá 16 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 en Twitter con el siguiente formato:

@everisCodeFEST #mireto4 <longitud> (indica tu nº de mesa)
Ejemplo: @everisCodeFEST #mireto4 6 (mesa 15)

- Si por algún motivo no tuvieras conectividad con Twitter, ponte en contacto con tu dinamizador asignado, que te ayudará a publicar el resultado.
- El ganador se decidirá según orden de entrada en Twitter con el hashtag indicado.

¡¡Suerte!!

{ code } { FEST }



SEVILLA, 3 DE NOVIEMBRE DE 2017

Reto 5

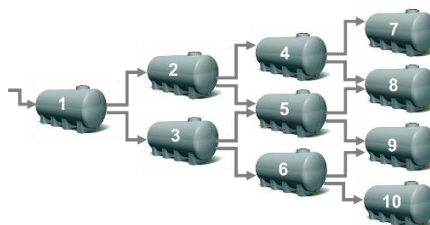
Canalización de depósitos industriales

TIEMPO DISPONIBLE: 60 MINUTOS

Un cliente del sector industrial está implementando un sistema de control en la distribución de depósitos de agua.

Tienen una red de canalizaciones que funciona de la siguiente manera:

- Todos los depósitos tiene la misma capacidad: 1.000 litros.
- El suministro de agua lo recibe un depósito principal (depósito 1).
- Cuando un depósito se llena, el exceso se vierte por partes iguales en 2 depósitos situados a continuación (depósitos 2 y 3). Y así sucesivamente con cada depósito, siguiendo el patrón jerárquico.
- Cada fila de depósitos tiene siempre un depósito más que la fila anterior.
- Los depósitos centrales tienen entrada de 2 depósitos diferentes como se indica en la figura (ver depósitos 5, 8 y 9):



Si los depósitos finales se llenaran verterían el exceso al alcantarillado. Por tanto todos los depósitos tendrán como máximo 1.000 litros.

El reto consiste en diseñar un algoritmo que permita calcular el volumen de agua alojado en un depósito concreto n , cuando tenemos una red de N depósitos que siguen el patrón indicado, después de que al depósito principal se le haya suministrado ya una cantidad C de agua, sabiendo que inicialmente todos los depósitos estaban vacíos.

Por ejemplo, para una red de 10 depósitos, cuando se suministran 3.400 litros al depósito principal:

- El depósito 1 se llena, quedando completo con 1.000 litros, desbordando los 2.400 restantes a los depósitos 2 y 3 por partes iguales.
- El depósito 2 y 3 se llenan también con 1.000 litros, desbordando 200 litros en cada uno hacia los depósitos 4, 5 y 6.
- El depósito 4 tendrá 100 litros, el depósito central 5 tendrá $100 + 100 = 200$ litros, y el depósito 6 tendrá 100 litros.
- Los depósitos restantes permanecerán vacíos.

Algunos ejemplos para que puedas probar tu algoritmo:

Número de depósitos (N)	Cantidad suministrada (C, en litros)	Depósito a calcular	Resultado (litros)
10	3.400	5	200
10	8.000	9	875
21	15.000	20	125

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:

Número de depósitos (N)	Cantidad suministrada (C, en litros)	Depósito a calcular	Resultado (litros)
210	90.318	164	¿?

- A la finalización del codeFEST se nombrará a la "escuela revelación 2017", como la escuela que haya conseguido más puntos entre sus alumnos. El ganador de este reto obtendrá 18 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 en Twitter con el siguiente formato:

@everisCodeFEST #mireto5 <resultado en litros> (indica tu nº de mesa)
Ejemplo: @everisCodeFEST #mireto5 200 (mesa 15)

- Si por algún motivo no tuvieras conectividad con Twitter, ponte en contacto con tu dinamizador asignado, que te ayudará a publicar el resultado.
- El ganador se decidirá según orden de entrada en Twitter con el hashtag indicado.

¡¡Suerte!!