

ESCUELA SUPERIOR POLITÉCNICA DEL LITORAL
FACULTAD DE INGENIERÍA EN ELECTRICIDAD Y COMPUTACIÓN
FUNDAMENTOS DE PROGRAMACIÓN
PRIMERA EVALUACIÓN - I TÉRMINO 2013-2014

Nombre: _____ **Matrícula:** _____

TEMA 1 (25 puntos)

Un **número Dudeney** es un entero positivo que es un cubo perfecto, de tal manera que la suma de sus dígitos decimales es igual a la raíz cúbica del número. **Por ejemplo**, los siguientes números:

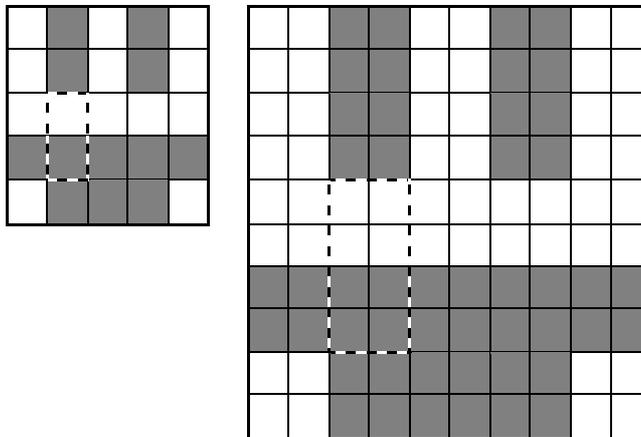
$$\begin{aligned} 1 &= 1; \text{ en donde } 1 = 1 \\ 512 &= 8 \times 8 \times 8; \text{ en donde } 8 = 5 + 1 + 2 \\ 5832 &= 18 \times 18 \times 18; \text{ en donde } 18 = 5 + 8 + 3 + 2 \\ 19683 &= 27 \times 27 \times 27; \text{ en donde } 27 = 1 + 9 + 6 + 8 + 3 \end{aligned}$$

Usted debe implementar en Scilab:

1. La función **[suma] = sumarDigitos(numero)**, la cual calculará la suma de los dígitos del número.
2. La función **[esDudeney] = esNumeroDudeney(numero)**, la cual retornará verdadero (%t) o falso (%f) en el caso que el número sea Dudeney o no. Use la función **sumarDigitos**.

TEMA 2 (25 puntos)

Al trabajar con imágenes, una operación común es la de agrandar la matriz que representa la imagen. Esto es, las dimensiones de la matriz crecen y los valores de celda original se copian a las adyacentes para llenar los espacios en blanco.



En la imagen pequeña:

- El pixel (3,2) se ve replicado en los pixeles (5,3), (5,4), (6,3) y (6,4) de la imagen grande.
- El pixel (4,2) se copia hacia los pixeles (7,3), (7,4), (8,3) y (8,4).

Escriba una función **matriz2 = agrandar(matriz1)**, que:

1. Recibe la imagen pequeña **matriz1**.
2. Calcula las dimensiones de **matriz1** y crea una matriz **matriz2** del *doble* de tamaño, llena de ceros inicialmente.
3. Recorre la matriz **matriz1** colocando las copias de sus pixeles en los lugares apropiados de **matriz2**.

TEMA 3 (50 puntos)

El juego “Expedición al Fondo del Mar” consiste en dirigir a un buzo por distintas partes del fondo del mar en su búsqueda de especímenes para investigación. Implemente la versión electrónica del juego con los siguientes requerimientos.

El buzo debe recorrer, con un tanque de 10 litros, 20 metros de distancia en un ambiente con temperaturas variantes y en ciertos metros de profundidad puede encontrarse con especímenes buenos o letales. La forma de replicar el descenso del buzo es con el uso, en cada turno, de un tanque de gas que le permite impulsarse al bajar en el agua. En cada turno se impulsa con el tanque y los metros que avanza dependen de la fórmula:

metros = $5 - T - (N * 0.3) - (G * 0.2)$, con el resultado redondeado para arriba.

- T= La temperatura en su posición (la cual es un número aleatorio entre -1 y -3).
- N= El número de especímenes que carga.
- G= El gas en su tanque (el cual disminuye 1 litro en cada turno).

En cada juego a lo largo de los 20 metros aparecerán 7 especímenes en posiciones aleatorias, seis buenos y uno letal. Si el buzo cae en un espécimen bueno lo recoge, si cae en uno letal pierde el juego. Gana el juego sólo si recorre los 20 metros con suficiente gas y ha recogido mínimo un espécimen para su investigación.

Asuma que posee la función **[M]=numerosSinRepetir()**, que devuelve un arreglo M con 7 números aleatorios entre 1 y 20 sin repetir. El primero representa el metro donde se encuentra el espécimen letal y el resto los metros de los especímenes buenos.

Usted debe implementar:

1. La función **[metros]=bajar(nEspecimenes,cantidadGas)** que recibe el número de especímenes que carga, la cantidad de gas en su tanque y retorna el número de metros que baja.
2. La función **[especimen]=obtenerEspecimen(metro, M)** que dado el metro y una matriz M generada por la función **numerosSinRepetir**, retorna 1 si cayó en un metro con un espécimen bueno, 0 si cayó en un metro con un espécimen letal y -1 si no hay nada en ese metro.
3. La función **mensajes(metros,especimen)** que dado el metro actual y espécimen (1 bueno, 0 letal y -1 sin espécimen), imprime en pantalla el metro donde se encuentra el buzo y además si ha encontrado o no un espécimen.
4. Un programa que simule el juego y **USE** las funciones implementadas anteriormente. En cada turno del jugador, se deberá mostrar por pantalla:
 - La cantidad de metros a bajar.
 - El metro donde se encuentra.
 - Si ha encontrado un espécimen.
 - La cantidad de oxígeno que le queda.
 - Si ha finalizado el juego.
 - Si ha ganado el juego.

* PUNTOS EXTRAS (5 puntos)

Implemente la función **numerosSinRepetir**.