

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

**Nombre:** \_\_\_\_\_ **Matrícula:** \_\_\_\_\_

**Tema 1 (25 puntos)**

Dos números son un **anagrama numérico** si uno de ellos puede ser construido a partir del reordenamiento de los dígitos del otro. **Por ejemplo:** 1971 - 1791 - 1197 - 1719 son anagramas numéricos porque un número podría resultar de la permutación de los dígitos del otro.

A usted se le solicita implementar en Scilab:

1. La función **[vectordelNumero] = convertirAVector (numero)**, la cual recibe un número como parámetro y retorna un vector que contiene sus dígitos.
2. La función **[esAnagramaNumerico] = verificarAnagramaNumerico (numero1, numero2)**, la cual recibe como parámetros dos números enteros positivos y retorna verdadero (%t) en caso que los números sean anagramas numéricos o falso (%f) en caso contrario. Use la función descrita en el literal anterior.

**Tema 2 (25 puntos)**

Un tablero de 9x9 contiene, en **posiciones aleatorias**, casilleros con minas, representadas por -1, y casilleros que están vacíos representados con 0. Para cada tablero, se debe determinar una matriz cuentaMinas con información sobre las minas, la cual se llena de la siguiente forma:

Las minas se copian en la misma posición. Los casilleros restantes se llenan con un número que resulta de contar el número de minas que existen en los casilleros adyacentes, excepto la casilla que contiene a la mina. Los casilleros adyacentes son aquellos que se encuentran en la posición siguiente ya sea vertical, horizontal o diagonalmente. A continuación, se muestra un **ejemplo** de un tablero y su correspondiente matriz cuentaMinas:

<b>tablero</b>										<b>cuentaMinas</b>									
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	1	1	0	0	0	0	0	0	0	0
-1	0	0	0	0	0	0	0	0	0	-1	2	1	0	0	0	0	0	0	0
0	-1	0	0	0	0	0	0	0	0	2	-1	1	0	0	1	1	1	0	0
0	0	0	0	0	0	-1	0	0	0	1	1	1	0	1	3	-1	2	0	0
0	0	0	0	0	-1	-1	0	0	0	0	0	0	0	1	-1	-1	2	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	2	2	1	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

A usted se le solicita implementar en Scilab:

1. La función **[cuentaMinas] = contarMinas (tablero)**, la cual recibe como parámetro un tablero con las minas y vacíos y retorna la matriz cuentaMinas.

### Tema 3 (50 puntos)

El mono-politécnico es un juego de mesa basado en el muy popular juego de monopolio. Esta versión del juego es entre dos jugadores que deben comprar propiedades. El juego se desarrolla en un tablero que consta de 40 casilleros, donde los casilleros 4, 12, 20 y 32 son casilleros especiales que generan un evento aleatorio y el resto representan las propiedades.

#### Eventos:

1. Ganar 500 dólares.
2. Ganar 1000 dólares.
3. Pagar 500 para ayudar a la reforestación de los bosques.
4. Pagar 700 dólares de impuestos al municipio.

Al iniciar el juego todos se encuentran en la posición uno (1) y cada jugador posee 4000 dólares. Si un jugador se queda sin dinero, pierde el juego y se declara ganador el otro jugador. El jugador que llegue al final del tablero se declara ganador del juego.

#### Acciones:

- Lanzar dos dados por turno y debe avanzar ese número de casillas.
- Decidir comprar o no la propiedad. El valor de la propiedad se calcula aplicando la siguiente fórmula:

$$\text{Valor de propiedad} = (\text{posición}/4+1)*100$$

- Si otro jugador cae en un casillero que ya ha sido ocupado, deberá pagar la renta. La renta de la propiedad se calcula aplicando la siguiente fórmula:

$$\text{Renta de propiedad} = (\text{posición}/4+1)*20$$

Utilice un arreglo M con 40 casilleros que representa el tablero y el contenido de cada casillero indicará el dueño del mismo (1 o 2).

Asuma que tiene la función **[estado] = verificarCasilla (pos,M)** que dado el arreglo M y la posición, retorna 0 si no tiene propietario, 1 si tiene propietario y 2 si la casilla representa un evento.

A usted se le solicita implementar en Scilab:

1. La función **[riqueza] = verificarEvento (pos,riqueza)** que dado la posición del jugador y el valor de su riqueza, retorna la nueva riqueza dependiendo del evento y muestra en pantalla el evento ocurrido **(5pts)**.
2. La función **[pos] = lanzarDado (pos)** que dada la posición del jugador y la bonificación, retorna la nueva posición del jugador y muestra un mensaje en pantalla de los casilleros que el jugador debe de avanzar **(8pts)**.
3. La función **[riqueza\_j1, riqueza\_j2] = pagarRenta (pos, riqueza\_j1, riqueza\_j2)** que dada la posición del jugador 1 y la riqueza que posee, devuelve la nueva riqueza del jugador 1 y jugador 2 **(4pts)**.
4. La función **[riqueza,M] = comprarPropiedad(pos, riqueza, M, jugador)** que dada la posición del jugador, la riqueza que posee, el arreglo M del tablero y el jugador, devuelve la nueva riqueza del jugador y asigna el propietario en el arreglo M **(5pts)**.
5. Un programa que simule el juego y **USE** las funciones implementadas anteriormente **(28pts)**.

En cada turno del jugador, se deberá de mostrar por pantalla:

- Su posición y su riqueza.
- El lanzamiento de los dados y la nueva posición.
- En la casilla que ha caído:
  - ❖ Si no tiene propietario, se debe mostrar un mensaje si va a comprar o no la propiedad.
  - ❖ Si tiene propietario, se debe mostrar el valor a pagar de la renta.
  - ❖ Si ha caído en un evento, se debe mostrar el evento ocurrido.