

Alumno: \_\_\_\_\_ C.I.: \_\_\_\_\_ Opción: \_\_\_\_\_

1. Realice el diseño algorítmico de las subrutinas para el siguiente problema: (5 pts.)  
Sudoku o Su Doku es el nombre del pasatiempo de moda por el que están apostando la mayoría de los periódicos. Las reglas son sencillas hay que rellenar las casillas del tablero de 9x9 con números del 1 al 9, de forma que no se repita ningún número en la misma línea, columna, o subcuadro de 3x3 que componen el Sudoku como se muestra en la figura 1.

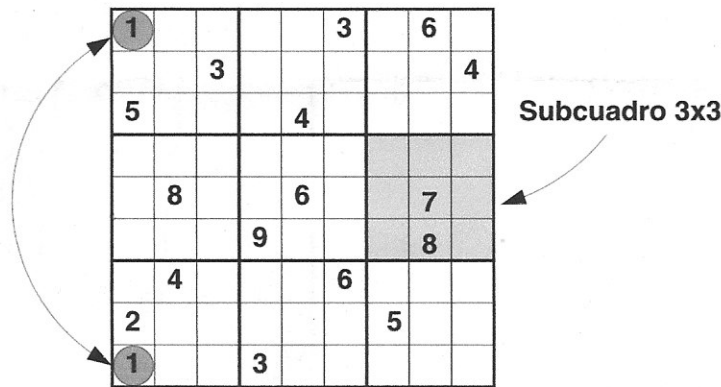


Figura 1: El jugador pierde por poner dos dígitos 1 en la misma fila. .

Realice el diseño algorítmico<sup>1</sup> de un subprograma que le permita verificar después de cada jugada si el dígito que se acaba de agregar al Sudoku no viola las reglas del juego.

2. Realice el diseño algorítmico que permita resolver el siguiente problema: (7 pts.)  
Dada dos matrices cuadradas  $M_1$  y  $M_2$  determinar si son matrices Inversas. Sabiendo que dos matrices se consideran inversas si al realizar el producto de ellas se obtiene la matriz identidad.  
LeerMatriz( $M \in$  Matriz,  $n \in \mathcal{Z}$ ), EscribirMatriz( $M \in$  Matriz,  $n \in \mathcal{Z}$ )
3. Realice el diseño algorítmico que permita resolver el siguiente problema: (8 pts.)  
Calcular el *Mínimo Común Múltiplo (mcm)* y el *Máximo Común Divisor (mcd)* de dos (2) números, haciendo uso de dos subrutinas donde:

- El *mcm* se define como el producto de los factores primos comunes y no comunes donde cada uno de ellos está elevado al mayor exponente.
- El *mcd* se define como el producto de los factores primos comunes donde cada uno de ellos está elevado al menor exponente que presente.

<sup>1</sup>recuerde que el diseño algorítmico debe incluir el análisis de entrada y salida, el proceso, el algoritmo y el análisis de variables; para cada uno de los programas.

S. Verifica Jugada (SGMatriz, f, c ∈ ℤ) ∈ ℤ

Valida ← cierto

Para i ← 1..9

Si ((S[i, c] = S[f, c]) ∧ ¬(i = f)) ∨  
((S[f, i] = S[f, c]) ∧ ¬(i = c))  
Valida ← falso

f1 ← 3 \* ((f-1) ÷ 3) + 1

c1 ← 3 \* ((c-1) ÷ 3) + 1

Para i ← f1..f1+2

Para j ← c1..c1+2

Si (S[i, j] = S[f, c]) ∧ ¬((i = f) ∧ (j = c))  
Valida ← falso

Devolver (valida)

V: valida ∈ ℤ

i, j, f1, c1 ∈ ℤ

T: Matriz : Areglo [1..9, 1..9] ∈ ℤ ∈ [1, 9]

E: n1, n2 ∈ ℤ

S: min, max ∈ ℤ

P: mcm : ∏ factores comunes y no comunes al mayor exp  
mcd : ∏ factores comunes al menor exp

A: Obtener (n1, n2)

min ← mcm(n1, n2)

max ← mcd(n1, n2)

Devolver (min, max)

V: min, max ∈ ℤ

S. mcd (a, b ∈ ℤ) ∈ ℤ

m ← 1, f ← 2

Repita

Mientras ((a ÷ f) = 0) ∧ ((b ÷ f) = 0))

m ← m \* f

a ← a ÷ f

b ← b ÷ f

f ← f + 1

Hasta ((f > a) ∨ (f > b))

Devolver (m)

V: m, f ∈ ℤ

E: M1, M2 ∈ Matriz

n ∈ ℤ n > 0 e ∈ ℝ e > 0

S: inversas ∈ ℤ

P: inversas = { cierto M1 \* M2 = I  
falso 0, c

A: Obtener (n, e)

Leer Matriz (M1, n)

Leer Matriz (M2, n)

Multiplica Matrices Cuadradas (M1, M2, I, n)

Inversas ← cierto

Para i ← 1..n

Para j ← 1..n

Si ((|I[i, j]| > e) ∧ (i ≠ j)) ∨  
((|I[i, j] - 1| > e) ∧ (i = j)))  
Inversas ← falso

Devolver (Inversas)

V: i, j ∈ ℤ

T: Matriz : Areglo [1..100, 1..100] ∈ ℝ

S Multiplica Matrices Cuadradas (A, B, C, n)

Para i ← 1..n

Para j ← 1..n

| C[i, j] ← 0

Para i ← 1..n

Para j ← 1..n

Para k ← 1..n

| C[i, j] ← C[i, j] + A[i, k] \* B[k, j]

V: i, j, k ∈ ℤ

S. mcm (a, b ∈ ℤ) ∈ ℤ

m ← 1, f ← 2

Repita

Mientras ((a ÷ f) = 0) ∨ ((b ÷ f) = 0))

m ← m \* f

Si ((a ÷ f) = 0) a ← a ÷ f

Si ((b ÷ f) = 0) b ← b ÷ f

f ← f + 1

Hasta ((f > a) ∧ (f > b))

Devolver (m)

V: m, f ∈ ℤ