

Comienzo y duración de cada ejercicio

Ejercicio 1: comienza a las 16:05 hs y tienen 20 minutos para hacerlo

Ejercicio 2: comienza a las 16:40 hs y tienen 50 minutos para hacerlo

Ejercicio 3: comienza a las 17:45 hs y tienen 50 minutos para hacerlo

Ejercicio 1 (3 puntos)

- a) Escribir una función que reciba un string y elimine del mismo todos los caracteres que representan un dígito
 - b) Escribir un programa completo (función main) que invoque a la función del punto a y provoque sí o sí un error de ejecución
-

Ejercicio 2 (3,5 puntos)

En esteganografía se ocultan valores en cada uno de los píxeles de una imagen. Para hacer este ocultamiento menos predecible se puede ocultar siguiendo una secuencia no correlativa de píxeles. En el esquema propuesto, se oculta entonces en los píxeles que correspondan según un salto elegido. Así, si el salto elegido es 2, se oculta en los píxeles de índices 0,2,4,6, etc. Cuando se llega al final de la cadena de píxeles, se sigue por los píxeles que faltan: los de índices 1, 3, 5, 7, etc.

Hacer una función que reciba un arreglo de píxeles y genere un nuevo arreglo de píxeles reordenado según el salto elegido para el ocultamiento de información.

Ejemplos:

Arreglo = [1,2,3,4,5,6,7,8,9]

salto = 2

Primero se sigue la secuencia 1, 3, 5, 7, 9 y al terminar se vuelve al inicio para seguir con 2, 4, 6, 8.

Nuevo = [1,3,5,7,9,2,4,6,8]

Arreglo = [1,2,3,4,5,6,7,8,9]

salto = 3

Primero se sigue la secuencia 1, 4, 7 y al terminar se vuelve al inicio para seguir con 2, 5, 8, y una vez más se vuelve al inicio y se toma 3,6,9.

Nuevo = [1,4,7,2,5,8,3,6,9]

Arreglo = [1,2,3,4,5,6,7,8]

salto = 3

Primero vendrá la secuencia 1, 4, 7 y al terminar sigue con 2, 5, 8, y finalmente 3,6.

Nuevo = [1,4,7,2,5,8,3,6]

Si el salto fuera cero o mayor a la dimensión, el vector de salida será igual al vector de entrada

Escribir una función con los siguientes parámetros:

1. vIn: vector de enteros con la cadena de píxeles original

2. dim: entero sin signo con la cantidad de elementos de vIn
 3. salto: entero sin signo que indica el salto
 4. vOut: vector de enteros donde almacenar los píxeles reordenados
-

Ejercicio 3 (3,5 puntos)

Escribir una función que compruebe si una solución parcial dada a un Sudoku es correcta (retorna 1) o inválida (retorna cero) sabiendo que:

- La función solamente recibe el tablero (una matriz de 9 x 9 enteros de un byte)
- Un tablero sudoku se compone de una matriz de 9 filas por 9 columnas. A su vez el tablero se subdivide en 9 subcuadrados de 9 casillas cada uno (3x3) NO SUPERPUESTOS (un número en una posición NO puede pertenecer a dos subcuadrados).
- El tablero puede estar incompleto. Si una casilla no está aún completa, tiene el valor cero
- Todas las casillas contienen un valor entre 0 y 9 (no se debe validar)
- Por fila, no hay valores repetidos (salvo el cero, si hay más de una celda incompleta en esa fila)
- Por columna, no hay valores repetidos (salvo el cero)
- Por subcuadrado, no hay valores repetidos (salvo el cero)

Ejemplos

Una matriz con todos ceros es válida

Si se recibe la siguiente matriz, la función retornará 1 (uno)

3	8	1	9	0	6	5	0	2
2	4	7	5	3	8	1	9	6
5	6	9	2	1	4	8	7	3
6	7	4	8	5	2	3	1	9
1	3	5	7	4	9	6	2	8
9	2	8	1	6	3	7	5	4
4	0	2	6	0	5	9	0	7
7	9	6	3	2	1	4	8	5
8	5	3	4	9	7	2	6	1

Si se recibe la siguiente matriz, la función retornará 0 (cero), porque por ejemplo la segunda fila tiene dos veces el 3

3	8	1	9	7	6	5	4	2
2	3	7	5	3	8	1	9	6
5	6	9	2	1	4	8	7	3
6	7	4	8	7	2	3	1	9
1	0	5	7	4	9	6	2	8
9	0	8	1	6	3	7	5	4
4	1	2	6	8	5	3	3	7
7	9	6	3	0	1	4	8	5
8	5	3	4	9	7	2	6	1

Si se recibe la siguiente matriz, la función retornará 0 (cero), porque el subcuadrado de arriba a la izquierda tiene dos veces el 3

0	0	0	9	7	6	5	4	0
2	3	0	5	0	8	1	9	6
5	6	3	0	0	0	0	0	0
6	7	4	8	0	2	0	1	9
1	0	5	7	4	9	6	2	0

9	0	8	1	6	3	7	5	4
4	1	2	6	8	5	0	0	0
7	9	6	3	0	1	4	8	5
8	5	0	4	9	7	2	6	1
