

## **MANEJO DE ARREGLOS STRING'S, VECTORES Y MATRICES (Autor de los ejercicios: Doctor Ricardo Moreno Laverde)**

**RECOMENDACIONES:** 1. Leer y entender el problema. 2. Saberlo hacer manualmente en papel y lápiz. 3. Construir un diseño de pantalla (Que me piden, que me dan y cómo quieren que se presente en pantalla. Para datos conocidos de entrada, datos conocidos de salida). 3. Construya el código en el lenguaje “ECMASCRIPT 6.0” (javascript 6.0 ) 4. Se recomienda **NO** utilizar chat GPT.

### **REQUISITOS O ESPECIFICACIONES**

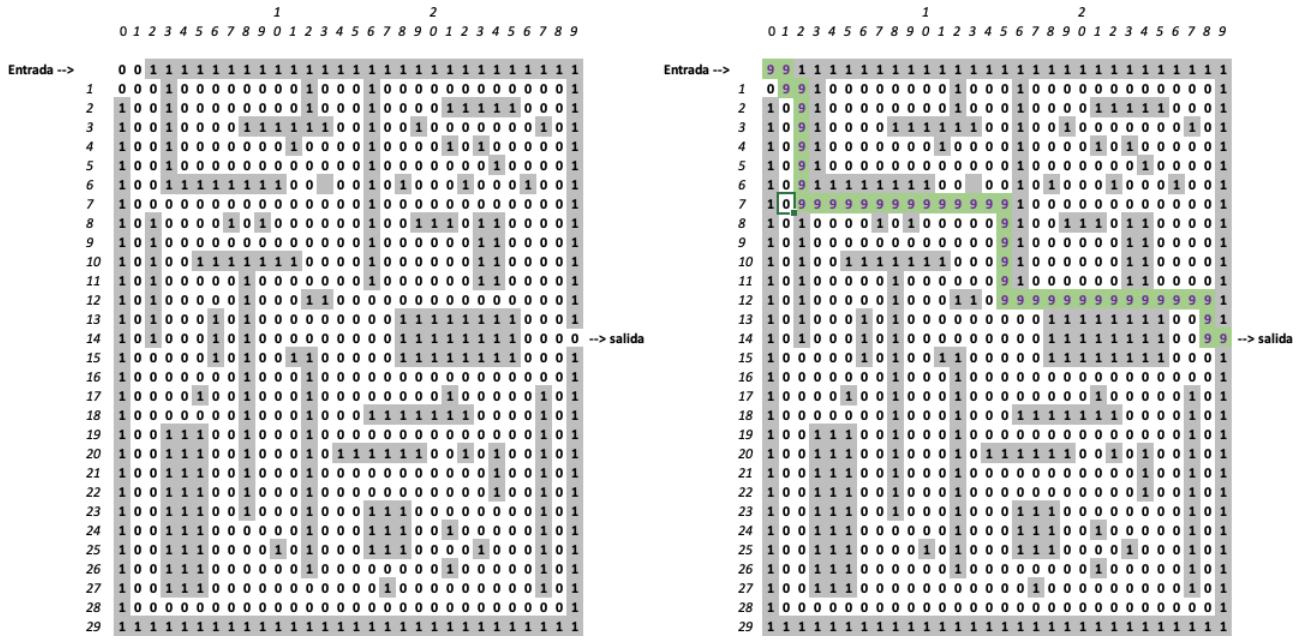
- Se requiere que cada punto tenga su propio programa. Si el ejercicio es de Factorial inverso, el archivo deberá llamarse FactorialInverso usando CamelCase Como nomenclatura
- El programa debe entregarse documentado y amable con el usuario. En los comentarios dentro del código, como documentación de la aplicación, que como mínimo debe tener:
  - Fecha y Hora de publicación
  - Versión de su código
  - Autores. Ing(c)
  - Nombre del lenguaje utilizado
  - Versión del lenguaje utilizado
  - Universidad Tecnológica de Pereira
  - Programa de Ingeniería de Sistemas y Computación
  - Un descriptivo de que hace el programa
  - -Para cada variable, haga un comentario de para que la usara dentro del código
  - -Solo haga lo que le piden, no agregue nada adicional.
  - -Salvedades si las hubiese. Vgr: Salvedad: Para valores fuera de este rango, no garantizamos los resultados.

**Todos los archivos deben subirse en el assigment del classroom en un directorio (folder) en una carpeta llamada “JavaScriptProyecto12” el Link del repositorio debe ser subido al Classroom con fecha máxima de entrega el 19 de Abril del 2024 antes de las 2:00 pm**

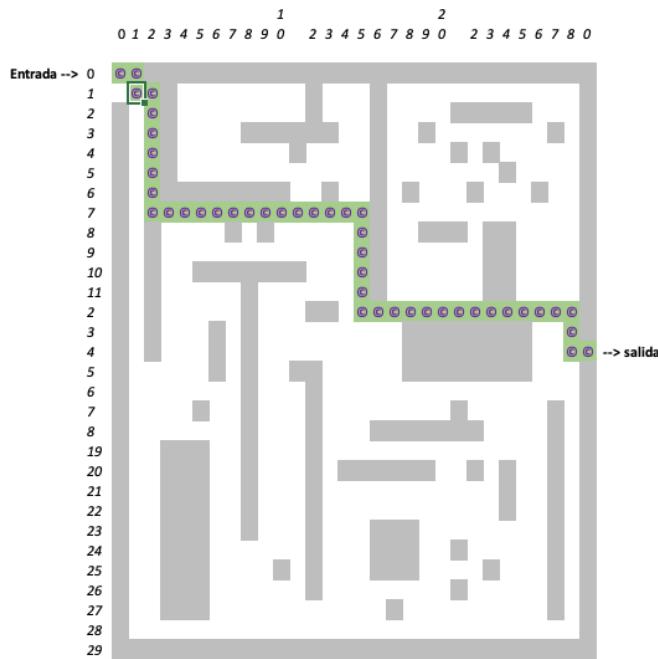
**Ejercicio 01. Laberinto.** Utilizando solo arrays, realizar un programa que permita simular un laberinto con una matriz de dimensiones 30\*30 , donde los pasos libres estarán representados por el carácter ASCII=32 (espacio en blanco = sp = space) y los muros por el carácter ASCII = █ = 219. Se generará aleatoriamente el laberinto y en todo caso la periferia estará siempre con el carácter ASCII = 219 = █. (0,2 hasta 0,29); (0,29 hasta 29,29) (29,29 hasta 29,0); (29,0 hasta 2,0), las coordenadas de entrada será la fila 0 y columna 0, y la salida se generará en forma aleatoria sobre la columna más a la derecha del arreglo (columna 29), sin tocar el muro de la periferia. Se pide:

1. Construir e imprimir el arreglo en forma clara y ordenada.

2. Construir una función que encuentre una(1) salida. Si no es posible, debe indicar que no tiene salida. La salida o camino de salida desde la posición fila 0 y columna 0, será indicado sobre la misma matriz con el carácter ASCII = 169 = ©. Si hay salida debe imprimir el laberinto con esa salida. Ejemplo:



Se debe presentar solo el laberinto de la siguiente forma:



**Ejercicio 02. Utilice solo vectores.** Una empresa marca todos sus productos con un código QR como el de la figura, que es un arreglo de 30x30 píxeles negros y blancos(string de 900 caracteres). Es posible detectar un producto falsificado comprobando una serie de características en el código QR.



Se pide escribir un aplicativo, para gestionar estos códigos de acuerdo con los siguientes pasos:

1. Declarar un arreglo de 30 x 30 y llenarlo con números unos(1) y ceros(0) en forma aleatoria, donde el uno(1) represente un pixel negro y el cero(0) un pixel blanco. En todo caso los píxeles de la periferia estarán siempre en blanco. (0,0 hasta 0,29); (0,29 hasta 29,29) (29,29 hasta 29,0); (29,0 hasta 0,0). Debe imprimir este arreglo con los códigos ASCII = 219 = █ que representa el pixel negro y el ASCII=32 (espacio en blanco = sp = space) para el pixel blanco.

2. Comprobar si el código QR tiene las características de los códigos QR asignados por la empresa, y en caso contrario, calcular el número de errores detectado.

Información adicional: Los códigos QR asignados por la empresa cumplen los siguientes requisitos:

- Los píxeles de la periferia están siempre en blanco.
- El código tiene reservados un cuadrado de 7x7 píxeles encajado en la parte más externa superior derecha(sin incluir la periferia, esto es no incluir fila 0 ni la columna 29), que debe estar formado por al menos 25 píxeles negros, con cualquier disposición.
- En la séptima fila(subíndice 6 de la figura), las columnas de la figura con subíndices 2, 4, y 25 son píxeles de color negro.

Para la verificación de sus resultados; imprima el código QR de las siguientes maneras (Como usted genero el arreglo de forma aleatoria, este puede ser solo un ejemplo):

Esta primera figura debe imprimirse



Esta figura debe imprimirse debajo de la primera

|0|1|2|3|4|5|6|7|8|9|0|1|2|3|4|5|6|7|8|9|0|1|2|3|4|5|6|7|8|9|  
| 0 |  
| 1 |  
| 2 |  
| 3 |  
| 4 |  
| 5 |  
| 6 |  
| 7 |  
| 8 |  
| 9 |  
| 10 |  
| 11 |  
| 12 |  
| 13 |  
| 14 |  
| 15 |  
| 16 |  
| 17 |  
| 18 |  
| 19 |  
| 20 |  
| 21 |  
| 22 |  
| 23 |  
| 24 |  
| 25 |  
| 26 |  
| 27 |  
| 28 |  
| 29 |  
+-----+  
|0|1|2|3|4|5|6|7|8|9|0|1|2|3|4|5|6|7|8|9|0|1|2|3|4|5|6|7|8|9|

Luego imprima en pantalla, los resultados de la siguiente manera: Requisitos:

- a) Los píxeles de la periferia están siempre en blanco
- b) El código tiene reservados un cuadrado de 7x7 píxeles encajado en la parte más externa superior derecha(sin incluir la periferia), que debe estar formado por al menos 25 píxeles negros, con cualquier disposición
- c) En la séptima fila(subíndice 6 de la figura), las columnas en la figura 2, 4, y 25 del arreglo son píxeles de color negro.

Cualquier producto que no cumpla los tres requisitos anteriores se considera falsificado. El número de errores se calculará teniendo en cuenta solo el último requisito (c), asignando un punto por cada píxel de los citados que no sea negro.

Condición a: Se cumple

Condición b: NO se cumple. Número de pixeles negros = ##

Condición c: NO se cumple.

Errores: 2 No es original

Para la condición b: Siempre debe aparecer el número de pixeles negros, se cumpla o no la condición.

**Ejercicio 03.** Se pide construir un programa, tal que:

1. Lea en una cadena de caracteres(string) una función polinómica racional del tipo ;

$$f(x) = A_1 X^1 + A_2 X^2 + \dots + A_n X^n + \dots + A_{n-1} X^{n-1} + A_n X^n + b$$

Donde:  $A_1, A_2 \dots A_n$  son los coeficientes y  $b$  el termino independiente. Ejemplos:  $f(x) = "348X^5 - 784X^2 + 6"$ .

$f(x) = "-10-X"$ .  $f(x) = "-3X + 5 + 4X^35"$ .

#### Consideraciones:

- En cualquier caso la ecuación, **debe leerse como una cadena de caracteres.**

- Los coeficientes son enteros positivos y/o negativos.

2. Construya un arreglo de tipo numerico, bidimensional de 30 filas x 2 columnas; donde, la primera almacene los coeficientes y en la segunda las potencias que acompañen ese término de esa ecuación

3. Imprima en pantalla los valores de este arreglo. Ejemplos:

Para el string “348X(-5)-6X-784X2+6”

Para el string “-10-10-34X-345”

Coeficiente	Potencia
348	-5
-6	1
-784	2
6	0

Coeficiente	Potencia
-10	0
-10	0
-34	1
-345	0

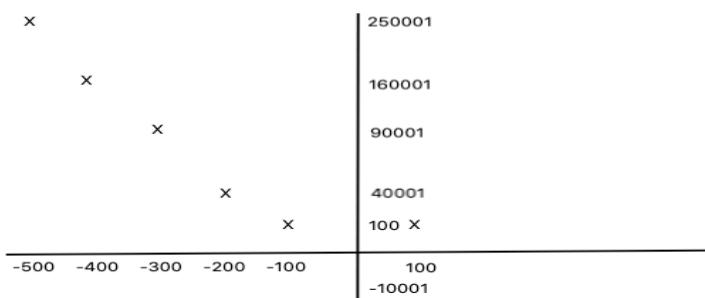
4. Leer desde teclado: Rango Inicio: Final: Incremento:

5. Construir la tabla de valores de  $f(x)$  para cada valor del rango entrado.

6. Graficar el polinomio, acorde a la tabla de valores; **en la pantalla en modo texto (24 filas x 80 columnas).** Ejemplo:  $Y = X^2 + 1$ , la cadena sería “ $X^2+1$ ”; si el rango es: inicio -500, final +100, intervalo +100, la tabla de valores sería:

X	Evaluación	Y
-500	$(-500)^2 + 1 = 250,001$	250,001
-400	$(-400)^2 + 1 = 160,001$	160,001
-300	$(-300)^2 + 1 = 90,001$	90,001
-200	$(-200)^2 + 1 = 40,001$	40,001
-100	$(-100)^2 + 1 = 10,001$	10,001
0	$(0)^2 + 1 = 1$	1
100	$(100)^2 + 1 = 10,001$	10,001

El gráfico sería en modo texto sería:



**Ejercicio 04.** Construir un programa de computador, tal que cree una pantalla (60 columnas x 30 filas) en modo texto y en forma aleatoria tal que aparezca lo siguiente en pantalla: *ejemplo:*

A. Esta primera figura debe imprimirse

B.  
Ahora, busque como conformar rectángulos, donde la letra "o" sea una esquina de ese rectángulo, dibújelos e imprima el área del menor de ellos (si los hay, para un rectángulo de mínimo dos(2) filas y/o dos(2) columnas).

Para este ejemplo el área más pequeña es  $6 \times 2$

Este es el lado superior tiene seis(6) columnas y la altura sería de dos(2) filas

C. Esta figura imprimala debajo de la primera

Otro ejemplo:

Para este ejemplo, tenemos tres(3) rectángulos: 1. Azul, 2. Rojo, 3.Amarillo. El área del más pequeño sería  $4 \times 3$

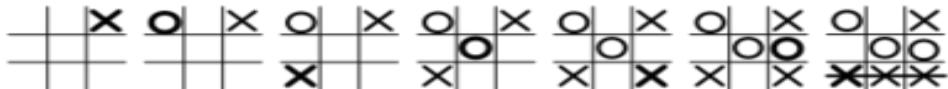
## Ejercicio 05.

Tomado y adaptado de: [https://es.wikipedia.org/wiki/Tres\\_en\\_1%C3%ADnea](https://es.wikipedia.org/wiki/Tres_en_1%C3%ADnea)

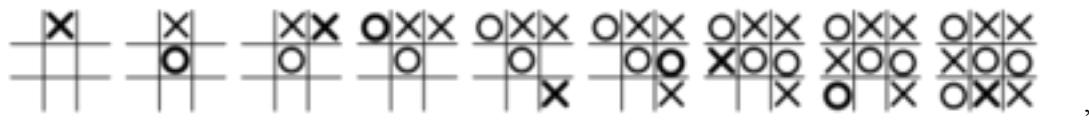
Fecha: 2021.04.07

“...El tres en línea, también conocido como ceros y cruces, tres en raya (España, Ecuador y Bolivia), cerito cruz (en Cuba), michi (en Perú), triqui (en Colombia), cuadritos, juego del gato, gato (en Chile, Costa Rica y México), tatetí (en Argentina, Paraguay y Uruguay), totito (en Guatemala), triqui traka, equis cero, tic-tac-toe (en Estados Unidos), o la vieja (en Venezuela), es un juego de lápiz y papel entre dos jugadores: O y X, que marcan los espacios de un tablero de  $3 \times 3$  alternadamente.

Ejemplo: Una partida ganada por el primer jugador que tiene las X:



Una partida que termina en empate:



Se pide hacer un programa tal que, permita jugar TRIQUI, tal que funcione así: Al jugador uno(1) se le asignara una “X” y al jugador dos(2) una “O”.

**Ejemplo:**

1 | 2 | 3  
---+---+---  
4 | 5 | 6  
---+---+---  
7 | 8 | 9

Jugador 1: Entre número casilla: 1

Al teclear uno(1) se colocara en esa casilla la X

X | 2 | 3  
---+---+---  
4 | 5 | 6  
---+---+---  
7 | 8 | 9

Jugador 2: Entre número casilla: 5

Al teclear cinco(5) se colocara en esa casilla la O

X | 2 | 3  
---+---+---  
4 | O | 6  
---+---+---  
7 | 8 | 9

Jugador 1: Entre número casilla: 5 Casilla ya jugada

Jugador 1: Entre número casilla:

- No se debe permitir que un jugador juegue una casilla ya jugada y si lo hace, debe permitírselle volver a entrar un número de casilla sin perder su turno.

El juego termina cuando se gane ó cuando se terminen de jugar todas las casillas.

**Ejercicio 06.** Se pide construir un algoritmo, tal que permita jugar a un(1) jugador, el juego de memoria llamado concéntrese (memorama) El juego consiste en “recordar” sobre la pantalla, donde están las parejas de letras, ocultas bajo el número del recuadro sobre el que este. Para nuestro ejercicio, serían cuatro(4) parejas. Para nuestro juego, tendríamos el siguiente diseño de pantalla:

JUEGO CONCENTRESE (MEMORAMA) Número de parejas = 0

1	2	3	4	5	6	7	8
---	---	---	---	---	---	---	---

Entre primera opción:

- Se debe utilizar cadenas de caracteres (string) y las funciones que permiten su manejo.
- Defina una cadena inicial, luego aplique un algoritmo que altere el orden aleatoriamente de la posición de los caracteres y juegue con estos. Ejemplo:

Cadena inicial

& & + + @ @ \* \*

& \* + & @ + \* @

Cadena modificada aleatoriamente

Supongamos que en memoria, tenemos, una cadena de caracteres(string) de ocho(8) posiciones, donde están las parejas, que el usuario debe “destapar”. Ejemplo:

&	*	+	&	@	+	*	@
1	2	3	4	5	6	7	8

Es decir:

- Pareja 01: es el recuadro 1 y el recuadro 4 Pareja 02: es el recuadro 2 y el recuadro 7 Pareja 03: es el recuadro 3 y el recuadro 6 Pareja 04: es el recuadro 5 y el recuadro 8

Si tuviéramos estos datos en la memoria del ordenador, juguemos con un ejemplo:

JUEGO CONCENTRESE (MEMORAMA) Número de parejas = 0 <table border="1"> <tr> <td>1</td><td>2</td><td>3</td><td>4</td><td>5</td><td>6</td><td>7</td><td>8</td></tr> </table> <p>Entre primera opción: 1</p>	1	2	3	4	5	6	7	8	Se destapa la casilla uno(1), mostrando la letra oculta.
1	2	3	4	5	6	7	8		
JUEGO CONCENTRESE (MEMORAMA) Número de parejas = 0 <table border="1"> <tr> <td>&amp;</td><td>2</td><td>3</td><td>4</td><td>5</td><td>6</td><td>7</td><td>8</td></tr> </table> <p>Entre segunda opción: 6</p>	&	2	3	4	5	6	7	8	Se destapa la casilla seis(6), mostrando la letra oculta.
&	2	3	4	5	6	7	8		
JUEGO CONCENTRESE (MEMORAMA) Número de parejas = 0 <table border="1"> <tr> <td>&amp;</td><td>2</td><td>3</td><td>4</td><td>5</td><td>+</td><td>7</td><td>8</td></tr> </table> <p>Desea continuar(s/n) : s</p>	&	2	3	4	5	+	7	8	Como las letras de las casillas destapadas no coinciden, no se completa la pareja y se ocultan nuevamente las letras a memorizar o recordar.
&	2	3	4	5	+	7	8		
JUEGO CONCENTRESE (MEMORAMA) Número de parejas = 0 <table border="1"> <tr> <td>1</td><td>2</td><td>3</td><td>4</td><td>5</td><td>6</td><td>7</td><td>8</td></tr> </table> <p>Entre primera opción:</p>	1	2	3	4	5	6	7	8	
1	2	3	4	5	6	7	8		

JUEGO CONCENTRESE (MEMORAMA) Número de parejas = 0 <table border="1"> <tr> <td>1</td><td>2</td><td>3</td><td>4</td><td>5</td><td>6</td><td>7</td><td>8</td></tr> </table> <p>Entre primera opción: 3</p>	1	2	3	4	5	6	7	8									
1	2	3	4	5	6	7	8										
JUEGO CONCENTRESE (MEMORAMA) Número de parejas = 0 <table border="1"> <tr> <td>1</td><td>2</td><td>+</td><td>4</td><td>5</td><td>6</td><td>7</td><td>8</td></tr> </table> <p>Entre segunda opción: 3 Casilla ya jugada Entre segunda opción: 6</p>	1	2	+	4	5	6	7	8									
1	2	+	4	5	6	7	8										
JUEGO CONCENTRESE (MEMORAMA) Número de parejas = 0 <table border="1"> <tr> <td>1</td><td>2</td><td>+</td><td>4</td><td>5</td><td>+</td><td>7</td><td>8</td></tr> </table> <p>Desea continuar(s/n) : s</p>	1	2	+	4	5	+	7	8	=1 <table border="1"> <tr> <td>1</td><td>2</td><td>+</td><td>4</td><td>5</td><td>+</td><td>7</td><td>8</td></tr> </table> <p>Entre primera opción:</p>	1	2	+	4	5	+	7	8
1	2	+	4	5	+	7	8										
1	2	+	4	5	+	7	8										
JUEGO CONCENTRESE (MEMORAMA) Número de parejas																	

Se destapa la casilla tres(1), mostrando la letra oculta.

Se destapa la casilla seis(6), mostrando la letra oculta.

Aquí al teclear como segunda opción la casilla tres(3) que ya fué jugada, debe advertirlo y permitirle entrar nuevamente la segunda opción.

Al haber coincidido que la letra oculta de la casilla tres(3) es igual al de la casilla seis(6), se incrementa el número de parejas, se dejan destapadas estas casilla y se sigue jugando.

El juego acaba cuando se completen las cuatro(4) parejas ó cuando a el jugador se le pregunte " Desea continuar(s/n):" y él teclee **n**

---

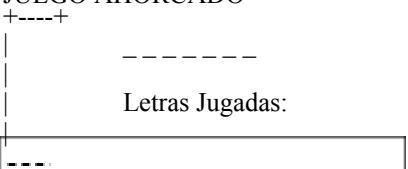
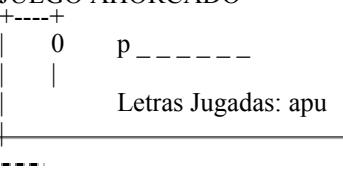
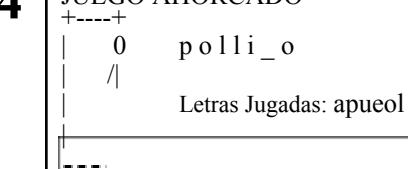
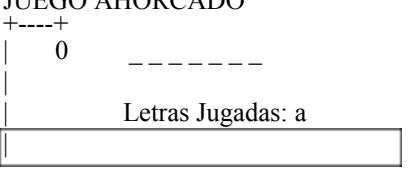
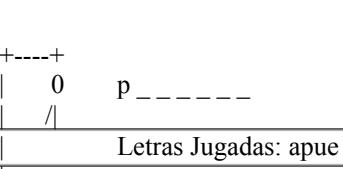
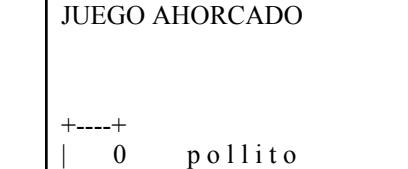
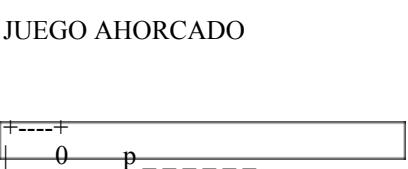
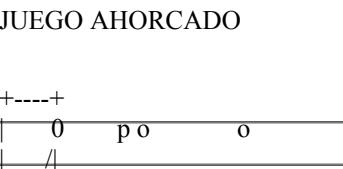
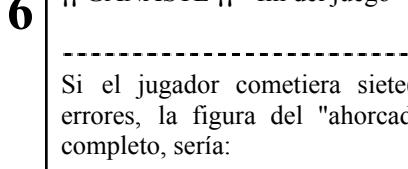
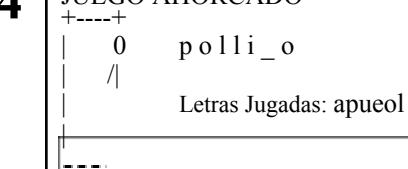
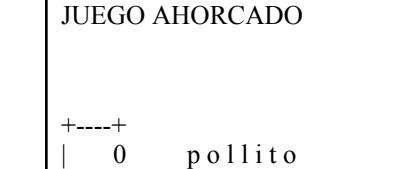
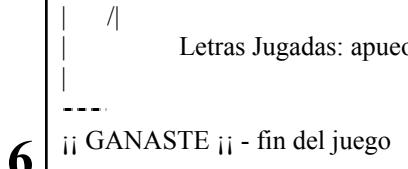
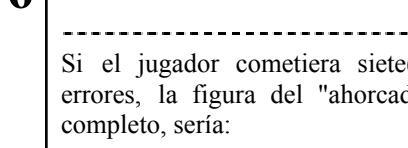
**Ejercicio 07.** Se pide hacer un programa tal que, permita jugar “Ahorcado”, y que funcione así: El juego consiste en que el jugador, debe adivinar una palabra oculta, que irá construyendo letra a letra, de tal forma que si la letra entrada no esta dentro de la palabra a adivinar, el jugador ira siendo “ahorcado” mostrando su cuerpo parte por parte. Para salvarlo es necesario adivinar una palabra, de la cual sólo se conoce su longitud. El jugador debe ir eligiendo letra por letra, de modo de ir completando la palabra. Si el jugador se equivoca en una letra, es decir, la letra seleccionada no pertenece a la palabra a adivinar, el personaje pierde alguna parte de su cuerpo (un brazo, una pierna, el tronco, etc). Se puede jugar hasta que el personaje pierda todo su cuerpo, el último resto de su trágica vida.

Requisitos:

- Se pide crear hasta diez(10) palabras, tal que el programa escoja aleatoriamente una de ellas, para que el jugador la adivine construyéndola letra por letra.
- Lea cada letra de la palabra, desde el teclado.
- El jugador debe adivinar las letras que pertenecen a la palabra y el programa debe terminar cuando todas las letras se hayan adivinado o bien se haya cometido un número establecido de desaciertos, es decir, el jugador ha muerto “ahorcado”.
- El código a nivel de comentario debe contener: El enunciado del problema, el autor, la fecha, un descriptivo de cada “identificador”(variable) a utilizar(utilice nombres nemotécnicos, que representan para que se utilizan).
- **Solo puede utilizar string's.** No puede utilizar arreglos, listas u otro temas no desarrollados en clase hasta el día de hoy.
- Nótese que las letras son entradas sin utilizar doble comilla.

Ejemplo: Supongamos que: a. Se tomen las siguientes palabras: 1- “ola”. 2-”pollo”.3-“murcielago”. 4-“universidad”. 5-“pereira”.

b. Que el número aleatorio generado sea el dos(2), esto es que la palabra elegida para este juego sería “pollo”

<b>JUEGO AHORCADO</b>  Letras Jugadas: <input type="text"/>	<b>1</b> <b>JUEGO AHORCADO</b>  Letras Jugadas: apu <input type="text"/>	<b>4</b> <b>JUEGO AHORCADO</b>  Letras Jugadas: apueol <input type="text"/>
<b>JUEGO AHORCADO</b>  Letras Jugadas: a <input type="text"/>	<b>2</b> <b>JUEGO AHORCADO</b>  Letras Jugadas: apue <input type="text"/>	<b>5</b> <b>JUEGO AHORCADO</b>  Letras Jugadas: apueolt <input type="text"/>
<b>JUEGO AHORCADO</b>  Letras Jugadas: ap <input type="text"/>	<b>3</b> <b>JUEGO AHORCADO</b>  Letras Jugadas: apue <input type="text"/>	<b>6</b> <b>JUEGO AHORCADO</b>  Letras Jugadas: apue <input type="text"/>
		<b>7</b> <b>JUEGO AHORCADO</b>  Letras Jugadas: apueol <input type="text"/> Entre una letra: t  <b>JUEGO AHORCADO</b>  Letras Jugadas: apueolt <input type="text"/> Entre una letra: i  <b>JUEGO AHORCADO</b>  Letras Jugadas: apueolt <input type="text"/> Entre una letra: GANASTE !! - fin del juego
		<b>8</b> Si el jugador cometiera siete(7) errores, la figura del "ahorcado" completo, seria: 

Entre una letra:  
u

L

l  
a

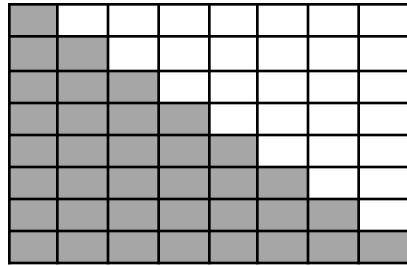
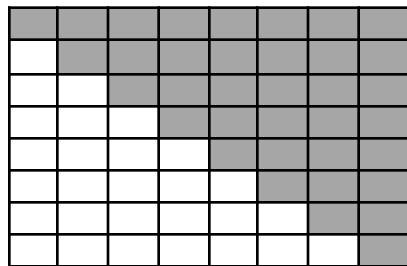
---

**Ejercicio 8:-** En un arreglo bidimensional de 8 x 8:

Elementos diagonal superior

Elementos diagonal inferior

0,0	0,1	0,2	0,3	0,4	0,5	0,6	0,7
1,0	1,1						
2,0		2,2					
3,0			3,3				
4,0				4,4			
5,0					5,5		
6,0						6,6	
7,0							7,7



Diagonal principal, elementos: 0,0 1,1 2,2 3,3 4,4 5,5 6,6 7,7

Construya un programa, utilizando funciones para cada caso, tal

que:

**1.-** Genere un arreglo de tipo float bidimensional de 8 x 8 - con números aleatorios entre 0.0 y 15.0 con un decimal e imprimirlo en pantalla de forma ordenada; por ejemplo:

7.4	8.2	9.1	10.9	0.0	3.3	5.1	6.1
2.0	10.0	1.6	15.0	10.1	1.4	1.8	3.4
3.0	2.9	1.9	12.9	9.1	4.5	2.3	2.7
5.4	3.1	2.0	13.3	8.2	2.6	3.5	3.6
4.3	8.3	1.7	1.6	4.6	8.5	4.7	8.2
6.1	6.0	2.5	2.8	12.9	11.9	3.8	1.5

Elementos aleatorios(generados)

Los colores son solo de referencia

7.0	3.5	1.9	1.9	11.0	12.0	14.0	1.7
8.2	2.9	13.0	14.7	5.7	1.8	12.6	1.9

**Construya solo funciones tipo flecha.**

**2.- Moda:** El valor que mas se repite en el arreglo e indique cuantas veces esta.

Ejemplo: Para esta secuencia de números aleatorios hay dos(2) modas

Imprimirla: Moda 8.2 esta 4 veces. Moda 1.9 esta 4 veces.

**3.- El promedio(a un decimal) de cada fila y de cada columna**

**4.- La (suma de la diagonal superior menos (-) la suma de la inferior) / promedio(a un decimal) del arreglo.**

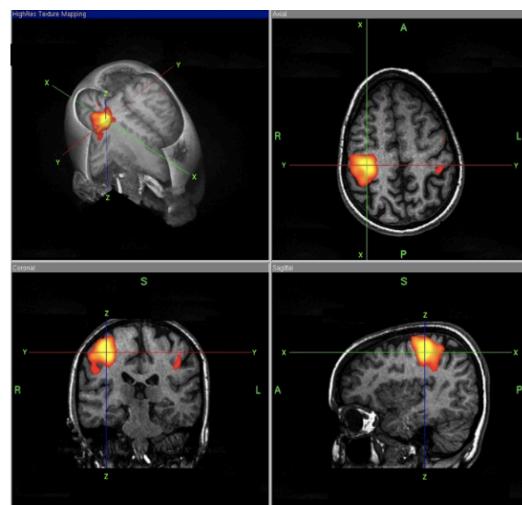
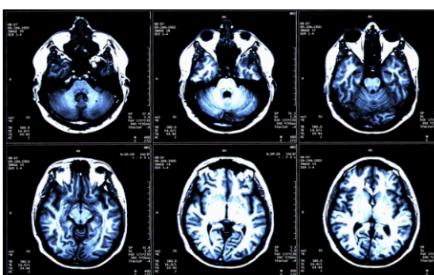
e.- Calcule e imprima en forma ordenada, un arreglo que contenga los factoriales de la parte entera de todos los valores generados en el enciso a.

**5.- Indique cuantas veces se encuentra el promedio.**

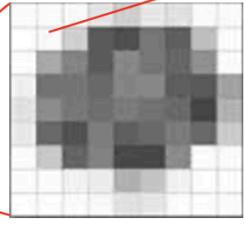
**6.- El promedio de los valores más altos de cada fila y cada columna**

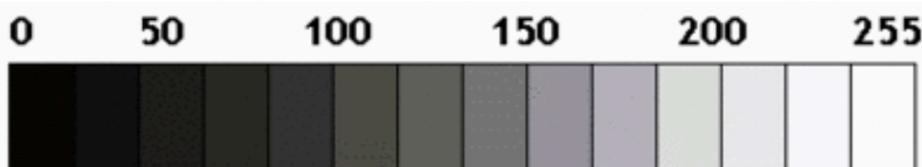
Realice pruebas de escritorio, para validar los resultados de su aplicación

### Ejercicio 9:

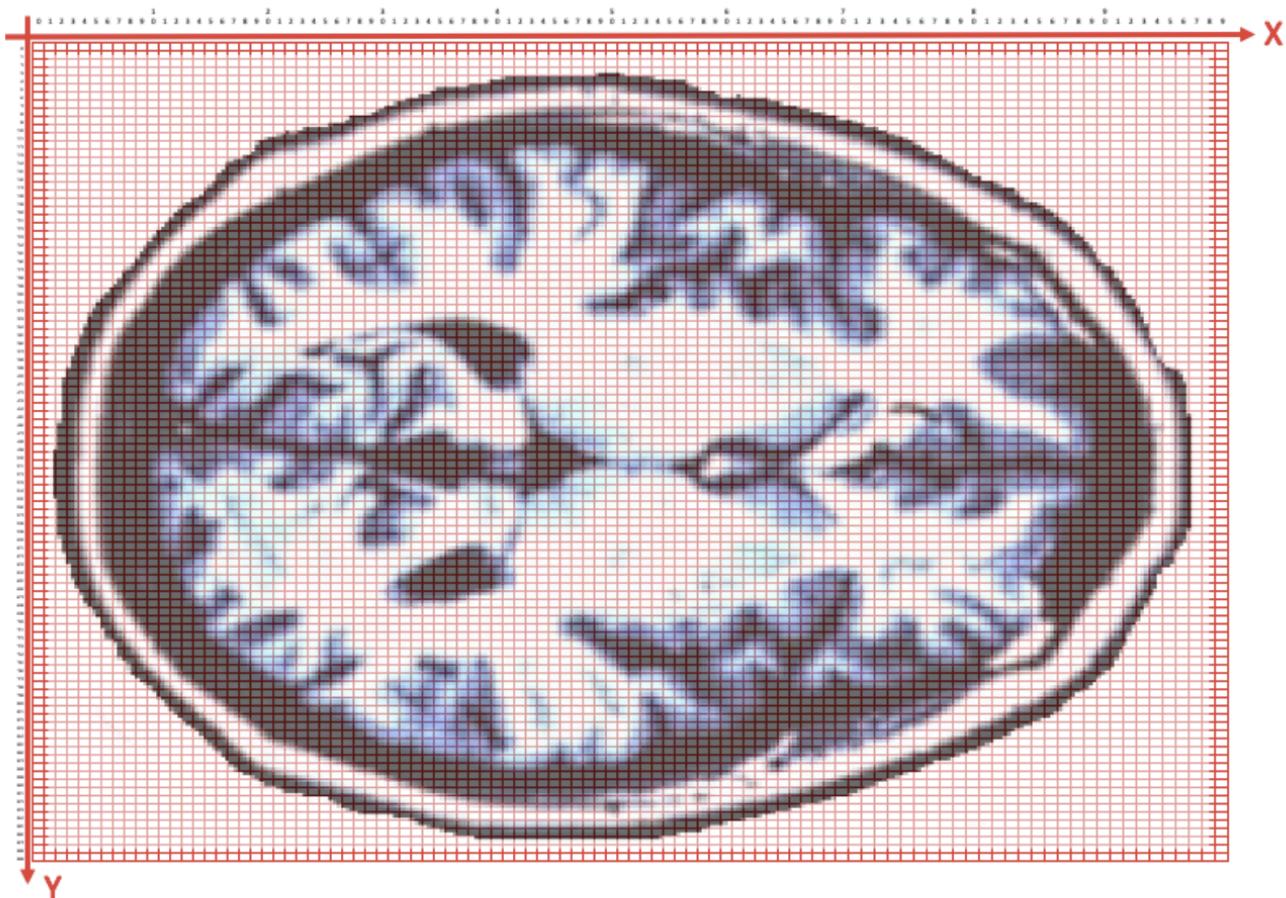


La resonancia magnética es una técnica de diagnóstico por imagen en la que se obtiene información de las características de diferentes puntos del cuerpo a partir de su respuesta a un campo magnético variable. Supongamos que tenemos una matriz tridimensional de puntos que representa una imagen adquirida por este método, en el que cada punto tiene un valor entre 0 y 255, correspondiente a un distinto nivel de gris.

IMAGEN	MUESTREO DIGITAL	CUANTIFICACION DE PIXELES																																																																																																				
		<table border="1"><tr><td>247</td><td>245</td><td>239</td><td>234</td><td>208</td><td>234</td><td>226</td><td>250</td><td>255</td><td></td></tr><tr><td>243</td><td>245</td><td>211</td><td>23</td><td>82</td><td>121</td><td>56</td><td>192</td><td>254</td><td></td></tr><tr><td>251</td><td>171</td><td>134</td><td>92</td><td>137</td><td>121</td><td>106</td><td>144</td><td>252</td><td></td></tr><tr><td>243</td><td>115</td><td>117</td><td>106</td><td>135</td><td>139</td><td>55</td><td>53</td><td>161</td><td></td></tr><tr><td>278</td><td>143</td><td>121</td><td>114</td><td>125</td><td>108</td><td>126</td><td>107</td><td>205</td><td></td></tr><tr><td>235</td><td>107</td><td>85</td><td>126</td><td>98</td><td>109</td><td>126</td><td>107</td><td>206</td><td></td></tr><tr><td>242</td><td>202</td><td>103</td><td>133</td><td>74</td><td>72</td><td>142</td><td>236</td><td>252</td><td></td></tr><tr><td>254</td><td>252</td><td>244</td><td>239</td><td>178</td><td>199</td><td>243</td><td>249</td><td>245</td><td></td></tr><tr><td>255</td><td>244</td><td>243</td><td>250</td><td>227</td><td>232</td><td>241</td><td>251</td><td>253</td><td></td></tr></table>											247	245	239	234	208	234	226	250	255		243	245	211	23	82	121	56	192	254		251	171	134	92	137	121	106	144	252		243	115	117	106	135	139	55	53	161		278	143	121	114	125	108	126	107	205		235	107	85	126	98	109	126	107	206		242	202	103	133	74	72	142	236	252		254	252	244	239	178	199	243	249	245		255	244	243	250	227	232	241	251	253	
247	245	239	234	208	234	226	250	255																																																																																														
243	245	211	23	82	121	56	192	254																																																																																														
251	171	134	92	137	121	106	144	252																																																																																														
243	115	117	106	135	139	55	53	161																																																																																														
278	143	121	114	125	108	126	107	205																																																																																														
235	107	85	126	98	109	126	107	206																																																																																														
242	202	103	133	74	72	142	236	252																																																																																														
254	252	244	239	178	199	243	249	245																																																																																														
255	244	243	250	227	232	241	251	253																																																																																														

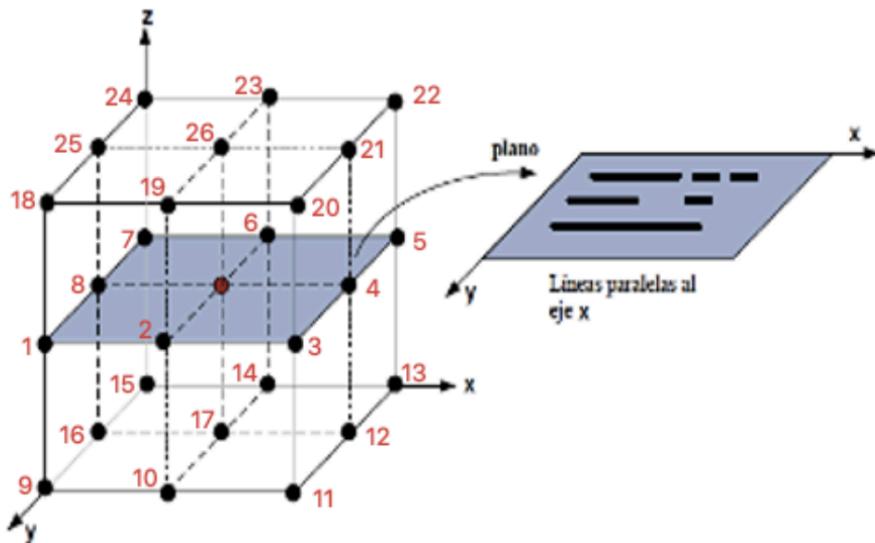


La imagen será como la de un cuaderno de 100 hojas, donde cada hoja tendrá una cuadrícula de 100 x 100 puntos, esto es que genera un arreglo de 100 planos (fotografías) en  $Z(0,1,2,3,4\dots 99)$

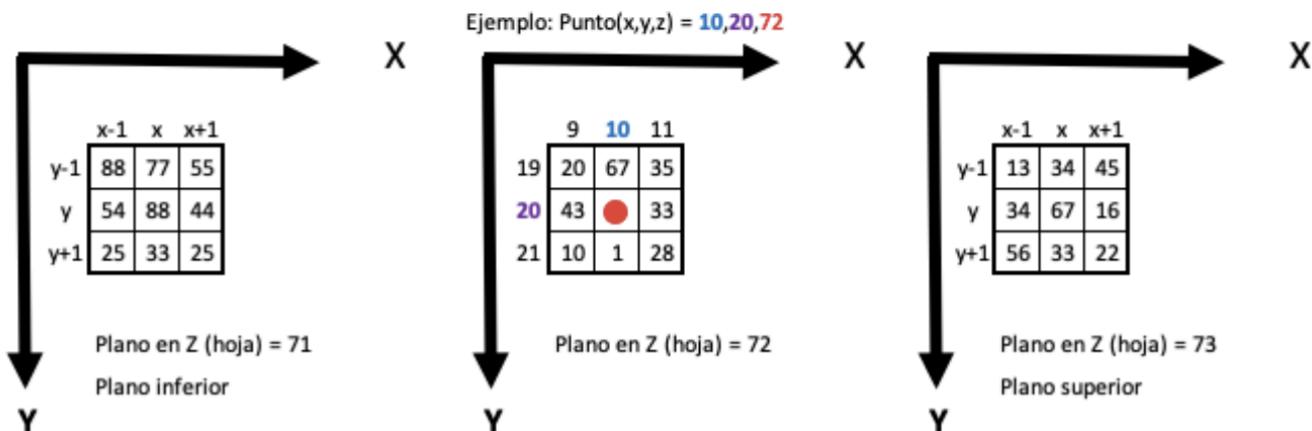


Se quiere escribir un programa que permita detectar aspectos en la imagen que pueden ser indicadores de una enfermedad. Como no tenemos los datos que generaría la máquina de “Resonancia magnética”, genere usted el arreglo con datos aleatorios entre [0 y 255] para probar sus funciones. Para ello se pide:

**1. Escribir una función** que, para un determinado punto en la imagen (identificado por sus coordenadas  $x, y, z$ ) , detecte si es sospechoso. Se consideran sospechosos aquellos puntos para los que TODOS los puntos adyacentes tengan un valor entre 20 y 40 (esto incluye los puntos pertenecientes al mismo plano, al plano inferior y al plano superior).



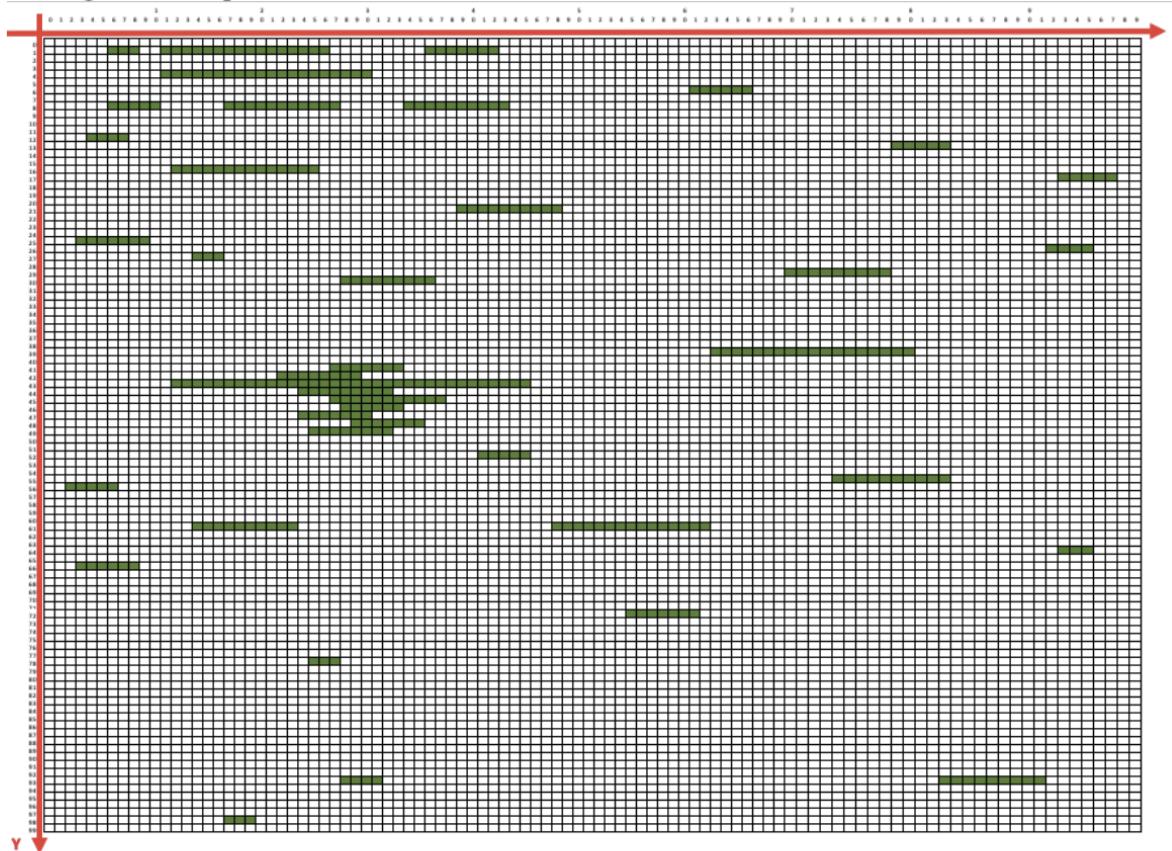
Nota: A la hora de analizar la imagen, no se analizarán los puntos de ninguno de los planos exteriores del cubo. Debe imprimirse las siguientes imágenes para este punto; así (ejemplo):



En este caso, para este ejemplo. Este punto NO ES SOSPECHOSO

2. Se pide escribir una función que, para un determinado plano de la imagen, busque líneas sospechosas. Se considera una línea sospechosa toda línea HORIZONTAL (paralelas al eje X) en la que haya al menos tres(3) puntos sospechosos consecutivos, para esto se pide que imprima el plano(fotografía, en modo gráfico); ejemplo:

Ejemplo: Fotografía en el plano Z=37



3. Escribir un programa que emplee las funciones anteriormente descritas para identificar en una imagen si hay alguna línea sospechosa y genere el siguiente informe(Ejemplo):

Plano en Z(Fotografía)	Líneas sospechosas x plano(photo)	Puntos sospechosos por plano(photo)
1	0	7
2	3	25
3	0	16
.....	....	.....
97	0	4
98	1	5

Los valores de todos los elementos de la matriz se supondrán ya cargados en memoria (no es necesario leerlos). En caso de que existan líneas sospechosas en la imagen, el programa mostrará en qué plano hay más líneas sospechosas (servirá para saber en qué lugar es mejor hacer una biopsia).

Ejemplo: Arreglo de 4 planos,  
cada uno de 100 x 100 puntos  
 $4 \times 100 \times 100$

