

Ejercicios Modelo de Segundo Global

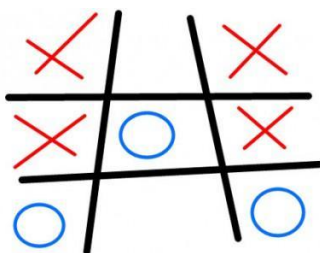
Reglas de Resolución:

1. Identifique con un comentario al inicio del código los datos de entrada y de salida.
2. En cada ejercicio es obligatorio el uso de al menos una variable n-dimensional.
3. Subdivida cada uno de los problemas de tal forma de utilizar al menos dos subprogramas.

1) Juego Ta-Te-Ti

Realizar un programa en PseInt que resuelva una partida del juego del “Ta-Te-Ti”. Un resumen de las reglas del juego son las siguientes:

1. El tablero es de 3x3 y cualquier cuadrícula podrá estar vacía u ocupada sólo por una cruz (X en rojo en el dibujo) o un círculo (O en azul en el dibujo).
2. Los jugadores juegan por turnos colocando una figura por vez en el tablero. No es importante si comienzan las cruces o los círculos.
3. Cada casillero es ocupado una vez por un jugador en la partida y no es posible cambiar dicho valor.
4. Hay un ganador cuando uno de los 2 jugadores forma Ta-Te-Ti, es decir, 3 valores del mismo tipo en horizontal, vertical o diagonal. Puede darse el caso que haya un empate o no haya ganador. No es necesario completar el tablero completo para saber si hay un ganador o no.



El programa debe permitir cargar los valores de los competidores, alternando los turnos, y debe determinar si algún jugador ha ganado la partida, o si ha habido un empate. En caso de completar el tablero y que no haya habido ningún ganador se debe mostrar un mensaje.

2) **"MiLigaOnLine"** es una página web que asiste a la gestión de torneos deportivos. En dicha página el usuario puede crear una liga asignándole un nombre y luego asociar los equipos. Una vez finalizada la creación de la liga se carga el resultado de todos los encuentros entre los equipos y se va actualizando el puntaje que obtiene cada equipo: goles a favor, goles en contra y la diferencia de goles. Por ejemplo, suponiendo una liga de 4 equipos, y luego de la carga de los resultados, se indica una tabla de posiciones así:

Nombre de Liga: Grupo D – Mundial 2018

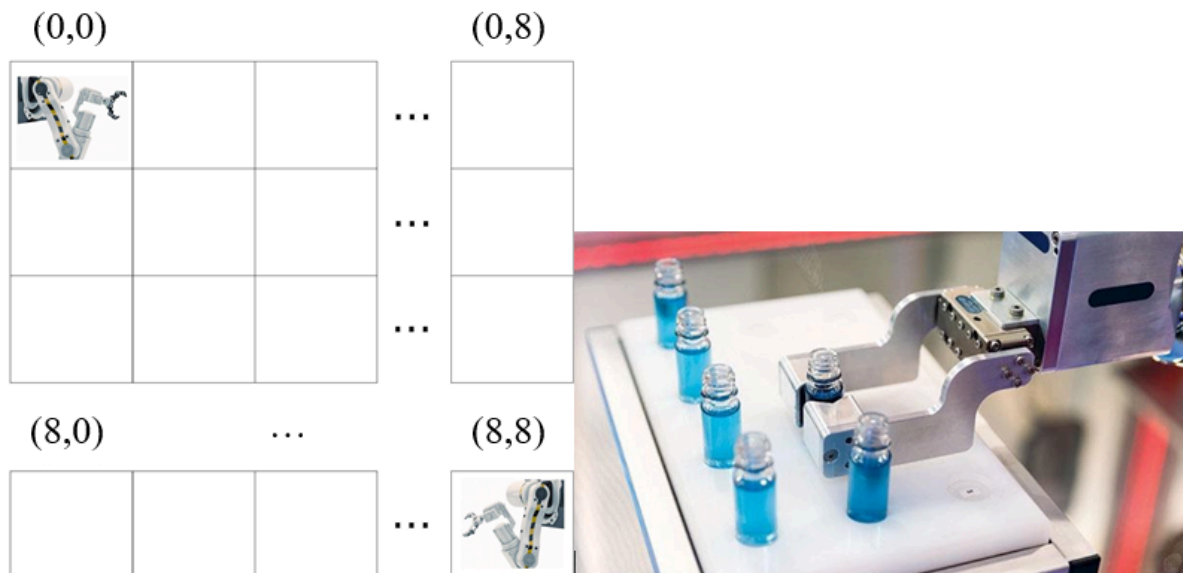
	<i>Puntos</i>	<i>Goles a Favor</i>	<i>Goles en Contra</i>	<i>Diferencia de Gol</i>
Argentina	9	6	1	+5
Croacia	4	4	3	+1
Islandia	3	2	2	0
Nigeria	1	1	7	-6

Sobre las reglas de la liga y funcionalidades del sistema:

- 1) El sistema no recuerda los resultados individuales de cada partido, sólo mantiene la tabla de posiciones. La tabla de posiciones se muestra luego de cargar cada encuentro.
- 2) Todos los equipos juegan contra todos los adversarios una sola vez. Los partidos siempre tienen un resultado.
- 3) El puntaje por encuentro es el siguiente: 3 puntos por ganar el encuentro, 1 punto por empate y 0 por derrota.
- 4) Al listar la diferencia de goles se debe identificar el signo positivo o negativo, salvo para el cero.

El sistema no ordena la tabla de posiciones.

3) “LabOrg 7000” es un robot mecánico para almacenar y manipular muestras de laboratorio en un tablero de 9x9 compartimentos que se identifican con una coordenada (X,Y). Para trabajar con las muestras el robot posee dos brazos robóticos que se encuentran ubicados en los extremos opuestos del tablero, ocupando las posiciones (0,0) y (8,8), respectivamente. A continuación se ilustra un ejemplo:



Este aparato posee una sencilla interfaz de usuario (UI) con las siguientes opciones:

1. Almacenar. Esta operación toma una muestra externa y la almacena en un compartimiento. Para ello, la UI solicita al usuario la posición donde almacenar la muestra, el brazo a utilizar (el 1 o el 2) y el nombre de la muestra a almacenar en ese compartimiento. Una vez finalizada la operación, el brazo seleccionado se queda “descansando” en esa posición. Finalmente la UI informa: “Se almacenó la “Nombre de Muestra” en la posición (X,Y)”.

2. Retirar. Esta operación toma una muestra almacenada y la extrae del aparato. Para realizar esta operación la UI solicita al usuario la posición desde la cual se retirará la muestra y el brazo que realizará el trabajo. Una vez finalizada la tarea, el brazo queda descansando en la posición original de cada brazo (ver información del manual más adelante). Una vez finalizada la tarea la UI informa “Se retiró la muestra “Nombre de Muestra” de la posición (X,Y)”

3. Mover. Esta operación toma una muestra de la posición (X,Y) y la mueve a la posición (M,N). Para ello, la UI solicita al usuario sólo la posición origen y destino. Luego, el “LabOrg 7000” en forma automática detecta cuál es el brazo que está más cercano a la posición origen, es decir, detecta el brazo que debe realizar menos movimientos y realiza la operación. Dado que los brazos sólo se mueven en dos ejes, el cálculo de la distancia es lineal. Por ejemplo, si un brazo está ubicado en la posición (6,1) y la muestra se encuentra en la posición (4,4), la distancia entre ese brazo y la muestra es igual a realizar 2 movimientos a la izquierda y 3 movimientos hacia abajo, es decir que la distancia es igual a 5 movimientos en total. El brazo queda descansando en la posición destino. Una vez finalizada la operación, la UI informa “El brazo B realizó el movimiento de la muestra “Nombre de Muestra” desde la posición (X,Y) hacia la posición (M,N)”.

4. Listar. Esta operación imprime un listado de las muestras y en qué posición se encuentra cada una. Por ejemplo:

“Nombre de Muestra 1” en la posición (X,Y)”



“Nombre de Muestra 2” en la posición (X,Y)”

...

“Nombre de Muestra N” en la posición (X,Y)”

5. Apagar. Apaga la máquina.

Sobre este modelo de robot el manual de usuario indica la siguiente información:

- Cada compartimento puede alojar sólo una muestra, por ende si se selecciona una posición ocupada la UI alerta: “La posición (X,Y) ya está ocupada por la muestra “Nombre de Muestra””
- Las posiciones (0,0) y (8,8) son ocupadas por los brazos 1 y 2, respectivamente, por lo tanto no son posiciones válidas para alojar las muestras.

Realice un programa que implemente la UI del “LabOrg 7000”.

4) “4 en línea” es un juego que consta de un tablero formado por siete columnas y seis filas. En una partida participan dos jugadores, uno con fichas blancas (B) y otro con fichas rojas (R). Inicialmente todas las posiciones del tablero están libres (L). Cada jugador coloca alternativamente una ficha en una columna. La ficha colocada cae por su propio peso hasta el fondo de la columna correspondiente (es decir, última fila libre de la columna). Por ejemplo, en la figura, si el jugador rojo (R) coloca una ficha en la columna 2, la ficha se coloca en la fila 4. La partida la gana el jugador que coloque primero cuatro de sus fichas contiguas en línea horizontal, vertical o diagonal. La partida queda en tablas (sin ganador), cuando ninguno de los jugadores es capaz de alinear cuatro fichas después de llenar el tablero.

	1	2	3	4	5	6	7
1	L	L	L	L	L	L	L
2	L	L	L	L	L	L	L
3	L	L	L	R	L	L	L
4	L	L	R	B	R	L	L
5	L	R	B	B	B	R	L
6	L	R	R	B	B	B	R

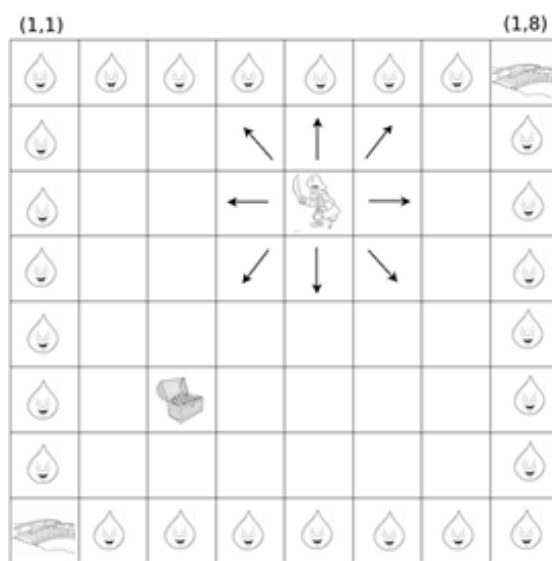
Un resumen de las reglas del juego es:

1. El tablero es de 6x7 y cualquier cuadrícula podrá estar libre (L) u ocupada sólo por una ficha blanca (círculo con la letra B en el dibujo) o una ficha roja (cuadrado con la letra R en el dibujo).
2. Los jugadores juegan por turnos colocando una figura por vez en el tablero. No es importante si comienzan las fichas blancas o las fichas rojas.
3. Cada casillero es ocupado una vez por la ficha de un jugador en la partida y no es posible cambiar dicho valor.
4. Hay un ganador cuando uno de los 2 jugadores forma “4 en línea”, es decir, 4 valores del mismo tipo de ficha en horizontal, vertical o diagonal. Puede darse el caso que haya un empate cuando ambos jugadores hayan jugado la misma cantidad de veces y formado “4 en línea”. No es necesario completar el tablero completo para saber si hay un ganador o no.

Realizar un algoritmo que resuelva una partida del juego “4 en línea” El algoritmo debe permitir cargar los valores de los competidores, alternando los turnos, y debe determinar si algún jugador ha ganado la partida, o si ha habido un empate. En caso de completar el tablero y que no haya habido ningún ganador se debe mostrar un mensaje.

5) El pirata **Pata de Palo** desea conseguir el gran tesoro sin morir ahogado. La isla, que consiste de un tablero cuadrado de 8x8 rodeado de agua y con puentes en dos esquinas, se puede representar así:

Para buscar el tesoro el pirata debe moverse por el tablero hacia una dirección que se determina de manera aleatoria (Norte, Sur, Este, Oeste o Diagonal) mediante el uso de la función *Azar()* de PSeInt. Un número aleatorio entre 1 y 5 decidirá qué dirección toma el pirata. Cuando la dirección establecida es “*Diagonal*” se debe determinar el sentido (Arriba-Izquierda, Arriba-Derecha, Abajo-Izquierda, o Abajo-Derecha) mediante otro número aleatorio entre 1 y 4, también generado a través de la función *Azar()*. Luego, una vez definida la dirección que seguirá el pirata, el usuario indica la cantidad de pasos que se moverá el pirata en esa dirección. Tener en cuenta que el pirata no puede dar más de 3 pasos seguidos en una misma dirección.



Para realizar la búsqueda del tesoro se deben tener en cuenta las siguientes reglas de juego:

- Cada vez que el pirata se mueve hacia alguno de los puentes en las esquinas, automáticamente se lo debe redireccionar al siguiente puente. Por ejemplo: si el pirata en uno de sus movimientos pasa por el puente de la posición (1,8) se lo debe reubicar automáticamente en la posición (8,1) y volver a sortear la dirección a seguir para el próximo movimiento.
- Cuando el pirata cae en una ubicación con agua se debe mostrar el mensaje “Ahogado!” y no se realizan más movimientos.
- Cuando el pirata encuentra el tesoro se muestra el mensaje “Eureka!” y finaliza el juego.
- La cantidad máxima de movimientos que puede realizar el pirata es igual a 20. Si al llegar a 20 movimientos el pirata no ha encontrado el tesoro, se debe mostrar el mensaje “Perdiste! El juego ha terminado”,

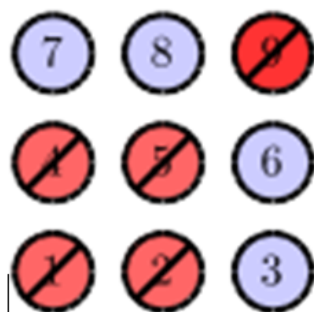
Realice un programa que permita al pirata “Pata de Palo” intentar encontrar el “Gran Tesoro”. Para ello, se debe crear el tablero y colocar al pirata Pata de Palo y el Gran Tesoro en casillas distintas determinadas de manera

aleatoria. El agua se ha de colocar en los bordes del tablero y los puentes en las esquinas indicadas en la figura. Una vez creado el tablero debe mostrarse por pantalla por única vez, y posteriormente, comienza una búsqueda a ciegas donde el usuario debe ir recordando las ubicaciones del pirata y el tesoro en el tablero hasta lograr encontrarlo.

6) ¡Pasa la calculadora! es un juego para dos jugadores que inicia con una calculadora recién encendida (mostrando el número 0 en el display). Cada jugador, de manera alterna, suma un número de un sólo dígito al valor acumulado hasta el momento de su turno, es decir que si el primer jugador suma el número 8 en su primer jugada, el display mostrará el número 8 al segundo jugador cuando le toque su turno.

Para ponerle un poco más de interés, en cada turno el jugador sólo puede utilizar los números situados en la misma fila o columna que el dígito marcado por su oponente en el turno anterior, pero no puede repetir el mismo número. Además, el número 0 no puede utilizarse nunca.

Por ejemplo, imagina que, durante una partida y luego de algunas jugadas, un jugador recibe la calculadora con el número 28 en el display, y además sabe que su oponente acaba de introducir el número 9. A partir de la disposición de los números de la calculadora, sabemos que en su turno el jugador únicamente podrá jugar con los números 3, 6, 7 y 8:



El jugador que tras sumar su número llega a un resultado mayor o igual a 31 pierde el juego y el mismo se da por finalizado. Por lo tanto, en el ejemplo anterior, todas las opciones posibles hacen que el valor sumado por el jugador llegue a 31 ($28 + 3 = 31$), y por tanto habrá, irremediablemente, perdido.

Realice un programa que implemente el juego ¡Pasa la calculadora!. Para ello, al iniciar el juego se deberá mostrar por pantalla la calculadora con los valores del 1 al 9 en la disposición que se muestra en la figura. Luego, en cada turno, se deberá mostrar al jugador el valor acumulado en el display hasta el momento, y además los posibles valores que puede ingresar luego de haber jugado su oponente, es decir que para el ejemplo anterior se deberá mostrar:

Turno: Jugador #1

Display = 28

Números posibles = 3, 6, 7 y 8

Una vez que el jugador haya ingresado su número el programa deberá mostrar el resultado de esa partida indicando si sigue “En juego” o “Pierde” (cuando alcanzó o superó el número 31). Si el jugador sigue en juego, se debe mostrar el display y los números posibles al siguiente jugador para que realice su jugada.

7) La “Batalla Naval Infantil” se juega en un tablero de 3 x 4, donde las filas se identifican de la A hasta la C y las columnas del 1 al 4. En el juego participan 2 contendientes: el defensor y el atacante. Dicho juego consiste en:

- i) El *defensor*, ubica 3 barcos: 1 barco doble (representado por un par de 2) y 2 barcos simples (representado por un 1 cada uno). Respecto de la ubicación de los barcos:
 - a) El barco doble debe ubicarse de tal forma que sus partes queden contiguas, ya sea horizontal o vertical, pero no es válido en forma oblicua.
 - b) Los barcos simples no poseen restricciones de ubicación.

Se ilustra a continuación un ejemplo de una ubicación posible:

	1	2	3	4
A	2			
B	2		1	
C				1

- ii) El *atacante*, indicando una coordenada del tablero (por ejemplo, la A3), debe intentar acertar a una embarcación de su contrincante. Luego de un ataque, se pueden presentar las siguientes situaciones:
 - a) Si el atacante no acierta, entonces el defensor informa “Agua!”.
 - b) Si el atacante acierta, suma el número del casillero al puntaje total del atacante. Por ejemplo, si al realizar el ataque le pega a una parte del barco doble, suma 2. Luego, el defensor informa “Hundido!” ó “Averiado!” en función de que el atacante haya destruido por completo (o no) el barco.

c) Si el atacante repite un disparo ya efectuado el defensor informa “Disparo repetido!”. Tener en cuenta que igual es un intento válido.

iii) Después de cada tiro, el defensor debe informar al atacante cuantos disparos le restan y cuantos puntos lleva sumados en su score.

iv) El juego termina cuando se cumple alguna de las siguientes situaciones:

- se han hundido todos los barcos. En este caso el atacante gana.
- cuando la cantidad de intentos de ataque es igual a 8 disparos. En este caso:
 - si el score supera el 80% del puntaje máximo el atacante gana.
 - de lo contrario gana el defensor.

Realice un algoritmo que implemente la lógica del juego, iniciando con la distribución de los barcos en el tablero por parte del defensor, y luego desarrollando la partida del atacante hasta la culminación del juego. El algoritmo debe indicar quién ganó.

8) PONIZOR

El barco carguero francés Ponziar traslada containers por el Atlántico. Por sus características sólo puede transportar tres tipos de containers: los de tamaño Large de 4 posiciones, los de tamaño Regular de 3 posiciones, y los Small que ocupan 2 posiciones.

El carguero organiza los containers horizontalmente en 5 posibles filas donde pueden ser ubicados, siendo la fila 3 la central, las filas 1 y 2 las que se ubican en proa, y las restantes en popa.

1 (proa)					
2 (proa)					
3 (central)					
4 (popa)					
5 (popa)					

Cada vez que llega un contenedor, el barco se debe organizar de tal manera que se maximice el espacio ocupado por las filas y que al mismo tiempo quede balanceado. Las reglas para lograrlo son sencillas:

1. Siempre se empieza cargando por la fila central.

2. Si el contenedor ingresado por el operador no entra en la fila actual entonces esa fila se considera llena y ya no se pueden cargar más contenedores en esa fila.
3. Cada vez que se llena una fila se avanza por las filas contiguas desde el centro hacia afuera, alternando entre proa y popa.
4. El barco queda listo para partir cuando no hay posibilidad de que se ingrese un contenedor más (sin necesidad de que el operador ingrese un contenedor nuevo), o cuando el contenedor ingresado por el operador no quepa en la última fila.

Se acompaña un ejemplo para entender cómo se van ubicando los contenedores:

Paso 1 - A - SMALL

Central 3,1

A	A			

Paso 2 - B - SMALL

Central 3,3

A	A	B	B	

Paso 3 - C - REGULAR

Proa 2,1

C	C	C		
A	A	B	B	

Paso 4 - D - LARGE

Popa 4,1

C	C	C		
A	A	B	B	
D	D	D	D	

Paso 5 - E - SMALL

Proa 1,1

E	E			
C	C	C		
A	A	B	B	
D	D	D	D	

Paso 4 - D - LARGE

Popa 5,1

E	E			
C	C	C		
A	A	B	B	
D	D	D	D	
F	F	F	F	

Realice un programa en PSeInt que ubique automáticamente los contenedores ingresados por un operador (indicando con una letra su tamaño L/R/S). En cada iteración el programa debe imprimir: la información del contenedor ingresado, la posición ocupada (central, proa o popa) con su coordenada inicial, y la disposición del contenedor dentro del barco en forma de matriz. El programa termina cuando el barco queda listo para partir, imprimiendo además, la cantidad total de contenedores cargados y la cantidad de espacios que quedaron libres.

9) JUEGO DEL 64

El juego del 64 consiste de un tablero de NxN celdas, siendo N un valor mayor o igual a 4 y menor o igual que 8. Inicialmente aparecen en él dos fichas con el número 2 en posiciones aleatorias no repetidas. A continuación se muestra un ejemplo para un tablero de 4x4:

	2		
		2	

El juego tiene las siguientes reglas:

1. El jugador puede elegir entre los siguientes 4 movimientos (arriba, abajo, derecha e izquierda).
2. Todas las fichas del tablero se desplazarán hacia el lado elegido por el jugador, teniendo en cuenta que cuando dos fichas colisionan, ambas se fusionan generando una ficha con la suma de los números de las fichas que han colisionado.
3. El desplazamiento se detiene por dos motivos:
 1. Porque se pudo realizar una fusión
 2. Porque no hay más espacios para desplazar, es decir, se llegó al borde de la matriz.
4. Cada vez que se realiza un movimiento se genera en el tablero una nueva ficha 2 en una posición aleatoria que esté libre luego de realizar los desplazamientos.

Por ejemplo, si luego de algunas jugadas el jugador decide mover hacia la derecha:

4	4	8	8
4		2	
			2
4			

DERECHA

	8	8	8
		6	
2			2
			4

y si en la siguiente jugada decide realizar un movimiento hacia abajo, el resultado será el siguiente:

	8	8	8
		6	
2			2
			4

ABAJO

		14	
	2		10
2	8		4

El juego termina cuando el jugador consiga una ficha con el número 64 (en este caso gana el juego), o bien, cuando se haya realizado un máximo de 20 movimientos (en cuyo caso el juego se declara desierto). Se debe mostrar por pantalla un mensaje con el resultado final del juego.