

PASATIEMPO IMAGEN ESCONDIDA

Se trata de un pasatiempo de lógica que forma una figura pixelada cuando se termina. El juego es similar al buscaminas, ya que se basa en una cuadrícula que esconde la imagen. El jugador de este solitario debe decidir qué cuadros se tienen que pintar de negro y cuáles no hasta que se tenga una visión completa de la figura.

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

Estos pasatiempos tienen un nivel avanzado en el que se tienen en cuenta la situación de varias pistas para decidir el color de una casilla, pero nosotros nos vamos a quedar con el nivel básico, en el que el pasatiempo se puede resolver mirando siempre los valores de una de las casillas. Se puede considerar una extensión de la práctica estos pasatiempos más complicados.

La figura anterior muestra el pasatiempo al comienzo, cuando aún no se ha decidido el color de ninguna casilla. Las dos siguientes muestran el puzzle en un estado intermedio durante el desarrollo, en el que ya se han decidido algunas casillas que permanecen desocupadas (marcadas con una X), y cuales pintadas de negro.

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

Finalmente, la figura siguiente muestra la figura final (¿un camión y un sol?) porque ya se ha decidido sobre si deben permanecer en blanco o negro todas las piezas.

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

El método de resolución se puede describir con un pasatiempo más pequeño. Supongamos que partimos del pasatiempo de la figura 1. Las pistas nos indican cuantas casillas se deben colorear en un recuadro formado por 9 cuadrillos: las casillas que rodean a una dada junto con dicha casilla. En el caso de los laterales, el número de casillas disminuye y en las esquinas aún más.

				1
	9			
	8	8		
				4
4		5		2

1

Un primer paso es identificar las posiciones de inicio, que son las casillas interiores con un 9, las laterales con un 6 y las de esquina con un 4. Se deben colorear todas esas casillas. Del mismo modo, las casillas interiores con un 0, indican que la casilla

debe permanecer sin colorear, así como las circundantes.

En la casilla (2,2) hay un 9 luego todos los cuadros circundantes deben ser negros, como se muestra en las figuras 2 y 3.



2



3

En la casilla (5,1) tenemos el mismo caso, pero aplicado a una esquina. Las esquinas solo delimitan 4 casillas, por lo que un 4 significa que las cuatro casillas afectadas se deben colorear (ver figuras 4 y 5).



4

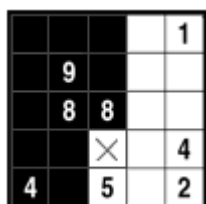


5

A continuación, las casillas que ya hemos decidido que son blancas o no, nos pueden ayudar con el resto de las pistas. Por ejemplo, en la posición (3,2) se indica que alrededor tiene que haber 8 casillas coloreadas (y ya las hay), por lo que se puede decidir que la restante debe quedar en blanco (ver figuras 6 y 7).



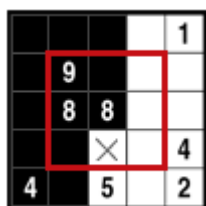
6



7

Y lo mismo ocurre con la pista de la posición (3,3). Dice que debe haber 8 casillas de 9 pintadas y ya se sabe cuál es la que no se tiene que pintar, por lo que se

puede colorear el resto de las casillas centradas en esa posición (figuras 8 y 9).



8



9

La pista de la posición (5,3) nos indica que de las seis casillas, deben estar coloreadas 5, y ya se sabe cuál es la que no, por lo que se pueden colorear el resto de las que aún no están decididas (figuras 10 y 11).



10

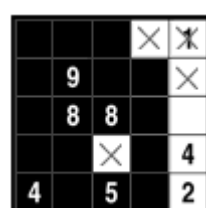


11

Lo mismo ocurre con la pista de la posición de esquina (1,5), que cubre un espacio en el que solo debe haber una casilla coloreada que ya está pintada, luego el resto deben permanecer en blanco (figuras 12 y 13).

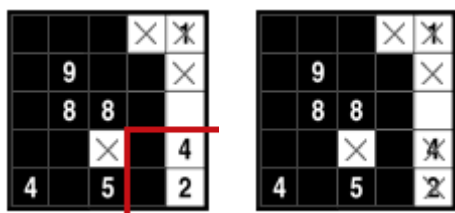


12



13

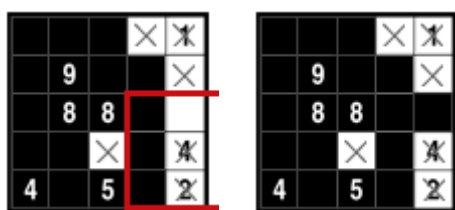
Finalmente, la pista de la posición (5,5), indica que en sus circundantes debe haber dos elementos pintados de negro, que ya están por lo que el resto deberá aparecer pintado de blanco (figuras 14 y 15).



14

15

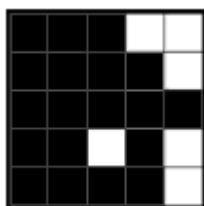
Y la pista de la posición (4,5) indica que a su alrededor debe haber cuatro casillas negras, solo hay tres y solo hay una posible sin decidir, por lo tanto, será negra.



16

17

Llegando a la imagen final, que representa una casa.



La práctica consiste en implementar varios subprogramas que se pueden utilizar en el desarrollo del juego. En el desarrollo se utilizarán la mayoría de los conceptos que se han visto en clase: Entrada/salida de datos. Se proporciona un tipo de datos para la implementación de los distintos subprogramas y se pide la realización de un programa que permita funcionar en dos modos: el modo automático que permite que sea el propio sistema descubra cuál es la solución del puzzle y el modo interactivo que permite al usuario indicar una coordenada y decidir si es correcta o no. Así mismo, tiene que dar la posibilidad de guardar el estado actual de la partida para poder retomarlo en un momento posterior.

Descripción del formato de entrada

Un juego se especifica mediante un par de enteros en una línea que representan el número de filas y de columnas que tiene la imagen. A continuación, tantas líneas como filas se indiquen, cada una con tantos caracteres como columnas. En las líneas estarán formadas por puntos o dígitos. El punto indica la ausencia de pista y un dígito entre 0 y 9 indica el valor de la pista en esa fila y esa columna del cuadro.

```
10 10
.....0.
.88.2.0...
5.8.....
.....2...2
1...456...
.0...79..6
...6..9..6
..668787.5
.4.666.6.4
.....3...
```



FASE 1 – Resolver el puzzle

Construir un programa que pregunta por el nombre de un fichero donde se especifica un juego (ver formato de entrada de datos), lo cargue en memoria en forma de una imagen sin resolver y una serie de pistas a utilizar para obtener la imagen. La imagen se guarda en una matriz bidimensional y las pistas sin resolver en una lista estática de pistas. Hasta conocer cómo se trabaja con ficheros en Ada, supón que los datos se escriben directamente en el teclado (para las pruebas puedes copiar el fichero de entrada y pegarlo con CTRL+V).

El programa, después debe ir resolviendo pistas y mostrando la evolución de la imagen a medida que se van resolviendo. Se resuelven las pistas hasta agotarlas o hasta que se detecte que no se puede avanzar en la resolución del puzzle. En el último caso, se indica la situación y el programa se debe detener y no quedarse ciclando indefinidamente.

Antes de acabar, el programa debe mostrar las pistas que se han resuelto y en el orden en el que se han hecho y que, se tienen que ir almacenando en una lista dinámica.

Fase 2 – Juego interactivo

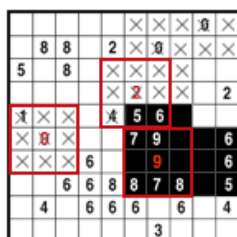
En este caso el juego es el mismo, pero en lugar de resolver el juego el programa ayuda al usuario a resolver el puzle, además de ofrecer la posibilidad de guardar el estado del juego para reanudarlo con posterioridad.

Para ayudar al usuario el programa debe mostrar el estado actual de la imagen y el de las pistas que aún no se han resuelto. El usuario elige una pista a resolver. El programa debe comprobar que la pista existe, debe comprobar que esa pista se puede resolver, y si es así, resolverla y mostrar la imagen y las pistas actualizadas.

Así mismo, el programa debe detectar si no hay más pistas que se puedan resolver y avisar al usuario.

Formato para guardar el estado de un juego y restaurarlo

Para guardar el estado de un juego, se guarda primero las dimensiones de la imagen: número de filas y columnas en sendos enteros en una línea, seguido del número de pistas que quedan por resolver. A continuación, se indica el número de pistas que quedan por resolver y tantos tríos de valores como pistas queden pendientes de resolver (en el texto se han tachado los que NO deberían aparecer por ya estar resueltos y se somborean las pistas alternas para facilitar la lectura). Finalmente, aparece el número de pistas que ya se han resuelto y la lista de las pistas resueltas en el orden en que se resolvieron.



10	10								
28									
2	2	8	2	3	8	2	5	2	3
4	6	2	4	10	2	5	1	1	5
5	7	6	6	6	7	7	4	6	7
8	4	6	8	5	8	8	6	7	8
8	10	5	9	2	4	9	4	6	9
9	8	6	9	10	4	10	7	3	
6	1	9	0	2	7	0	6	2	0
6	7	7	9				6	7	9

Laboratorio10

Se entrega la declaración de varios tipos de datos a utilizar y la especificación de varios subprogramas que se deben utilizar para resolver el problema. No es necesario implementar todos los subprogramas especificados, tan solo los que se vayan a utilizar.

El paquete no está probado, por lo que es posible que alguno de los elementos especificados sea erróneo. Si encuentras algún tipo de error, avisa a través del foro de la práctica, para que todos los que están haciendo la práctica estén al tanto del mismo.

Si quieres, puedes usar paquetes especiales para mejorar la entrada/salida del programa (como nt_console, que ofrece la posibilidad de controlar dónde se va a escribir a continuación o cambiar los colores del texto).

Si necesitas hacer subprogramas propios, te recomiendo crear un paquete de uso personal, si es que son auxiliares para el programa del juego. Si los programas son auxiliares para los programas del laboratorio10, los puedes añadir en el fichero laboratorio10.adb (no en el laboratorio10.ads). De esta forma, tus subprogramas los tendrán disponibles, pero los que usen el paquete no los podrán utilizar.