

# NAVEGACIÓN ESTELAR

La nave rebelde, Halcón Millonario, del mercenario Juan Solo ha sufrido graves daños en el último encuentro con los destructores imperiales. Para ser reparada debe llegar al planeta T evitando nuevos enfrentamientos con los destructores. Sin embargo, una avería en los sistemas de navegación de la nave hace que sólo pueda orientarse navegando en línea recta hacia una estrella E, y evitando tropezarse con los destructores.

## Objetivo

Dado un mapa de situación del Halcón Millonario (H), del planeta (T), de las estrellas y de los destructores, realizar un programa que permita calcular un camino para llegar al planeta T evitando los destructores D, **con el menor número posible de cambios de dirección**. El movimiento de la nave deberá ser en vertical, horizontal o diagonal, siempre en línea en dirección a una estrella E, y evitando tropezarse con los destructores.

Para simplificar el problema, hemos supuesto que todo el movimiento se realiza dentro de un mismo plano. El cuadrante dentro del que se realiza el movimiento se considera dividido en casillas (la cuadrícula nunca tendrá más de 30 filas ni 30 columnas y la ruta tendrá que realizarse sin salirse de ella). Cada impulso de motor permite trasladar la nave de una de estas casillas en la dirección de cualquiera de las 8 celdas que la rodean (la escasa resolución del sistema manual de control no permite realizar cambios de dirección más precisos). Pasar por una casilla ocupada por un destructor o una estrella supondría la destrucción inmediata del Halcón Millonario.

En el siguiente ejemplo puede verse el camino (en sombreado) que tendría que recorrer desde H hasta T en una supuesta situación. En esta ruta hay 2 cambios de dirección

			E				E		
				T					
		D	D	D					E
					D			D	
					E				
	E								
		D				D			
				H					

## Entrada

El programa deberá leer los datos del mapa desde un fichero de texto (ASCII) de nombre "ESTELA.IN". Este fichero tendrá la siguiente estructura:

- 1 En la primera línea contendrá el tamaño en filas (m) y columnas (n) del mapa separados por un espacio en blanco. ( $1 < m \leq 30$ ,  $1 < n \leq 30$ ).
- 2 En la segunda línea contendrá la posición X (columna) y la Y (fila) de la nave H. La coordenada superior izquierda del mapa corresponderá a la posición X=1; Y=1.

- 3 En la tercera línea aparecerán las coordenadas del planeta T.
- 4 En la cuarta línea contendrá el número de estrellas en el mapa.
- 5 A continuación aparecerá una línea por cada estrella con sus coordenadas X e Y.
- 6 Seguidamente, el fichero contendrá una línea con el número de destructores.
- 7 Finalmente, el fichero contendrá una línea por cada destructor con sus coordenadas X e Y en el mapa.

Naturalmente, todas las coordenadas X, Y indicadas cumplirán la condición ( $1 \leq X \leq n$ ,  $1 \leq Y \leq m$ ).

El archivo de entrada que correspondería a la situación dibujada arriba, puede verse en el ejemplo al final del enunciado.

### Salida

El programa deberá crear un fichero "ESTELA.OUT" con las coordenadas de todas las casillas por donde pasará la nave en su camino hacia T. Este camino deberá comenzar con la posición en que se encuentra la nave H y terminar con las coordenadas del planeta T.

En caso de que no haya ningún camino posible desde H hasta T, el fichero de salida deberá contener una única línea con el texto "NAVE PERDIDA".

El archivo de salida que correspondería a la solución dibujada arriba, puede verse en el siguiente ejemplo.

### Ejemplo:

ESTELA.IN	ESTELA.OUT
9 10¶	5 9¶
5 9¶	6 8¶
5 2¶	7 7¶
5¶	8 6¶
4 1¶	8 5¶
8 1¶	7 4¶
10 4¶	6 3¶
6 6¶	5 2¶
2 7¶	
7¶	
3 4¶	
4 4¶	
5 4¶	
6 5¶	
9 5¶	
3 8¶	
7 8¶	

NOTA: El carácter ¶ representa el salto de línea en los archivos de ejemplo.