

---

**Hundir la flota****P61920\_es**

---

¿Quien no ha jugado nunca a los barquitos con el compañero de al lado durante uno de esos concursos de programación en los que no sale nada? La OIE, pese a lo que pueda parecer, es una competición seria: vamos a darte la oportunidad de jugar a los barquitos... contra el ordenador.

En este problema tu programa jugará al hundir la flota. A diferencia de otros problemas, la entrada que tu programa recibirá será *interactiva*: tu programa escribirá una casilla donde atacar, y a cambio recibirá una respuesta.

En una partida de hundir la flota, hay un mapa rectangular de  $F$  por  $C$  casillas donde se distribuyen aleatoriamente barcos de distintos tamaños. Un barco es una fila o columna de  $t$  casillas consecutivas, donde  $t$  va desde 1 (guardacostas) hasta 5 (portaaviones). Dos casillas de barcos distintos nunca estarán en contacto compartiendo una arista (sí pueden estar en contacto compartiendo un punto, es decir, situadas diagonalmente una respecto la otra). Tu programa sabe cuántos barcos de cada tipo hay, pero no sabe dónde están colocados los barcos. Tu objetivo es hundirlos todos. Para hundir un barco es necesario dañar todas sus casillas.

En cada turno, tu programa tiene derecho a realizar un ataque sobre cualquiera de las casillas, escribiendo las coordenadas de la misma. A cambio, recibirá una de las tres respuestas posibles por su entrada: AGUA (si la casilla está vacía), TOCADO (si la casilla está ocupada por un barco al que todavía quedan casillas no tocadas) o HUNDIDO (si la casilla está ocupada por un barco cuyas casillas han sido todas tocadas). Por ejemplo: si hundes un barco de  $n$  casillas con  $n$  ataques, tu programa recibirá  $n - 1$  respuestas de TOCADO y una respuesta de HUNDIDO; si volvieras a atacar cualquiera de las casillas del mismo barco seguirías recibiendo el mensaje de HUNDIDO.

**Entrada**

Existen varios tableros de pruebas, todos ellos con distintos tamaños y números de barcos. Un juego de pruebas empieza con el número  $N$  de tableros.

Por cada tablero, deberás leer dos números  $F$  y  $C$  en una línea (el número de filas y columnas del tablero) seguido de una segunda línea con exactamente 5 números: las cantidades  $c_i$  de barcos de tamaño  $i$ , para  $i$  de 1 hasta 5. A partir de este momento la entrada que recibas dependerá de tus ataques. Cuando tu programa haya hundido todos los barcos, deberá escribir `FIN` por su salida, y empezar a leer el siguiente tablero. Cuando haya finalizado el número de tableros  $N$ , tu programa deberá acabar.

**Salida**

Escribe cada ataque en una línea, con las coordenadas  $i$  y  $j$  del ataque, separadas por un espacio, cumpliéndose que  $0 \leq i < F$  y  $0 \leq j < C$ . Escribe `FIN`, también en una línea, cuando detectes que no hay más barcos.

**Observación**

Tendrás a tu disposición un simulador y varios tableros de ejemplo para poder probar tú mismo tu programa. También te ofrecemos un ejemplo de jugador extremadamente tonto

(`tonto.cc`) para que tengas un ejemplo de código. Puedes basarte en el ejemplo para escribir tu programa. Para probar un ejecutable `jug.x` con los tableros que te damos, escribe `./prueba jug.x`. Recibirás como respuesta el número de turnos que tarda tu programa en escribir `FIN` para cada caso de pruebas. Si tu programa ha escrito `FIN` antes de haber hundido todos los barcos, o si ha escrito una casilla fuera del tablero, aparecerá `WA` (respuesta errónea) en el simulador. Comprueba que tu programa no tarda más de 10 segundos en resolver los casos de pruebas.

### Puntuación:

El sistema de puntuación de este ejercicio es distinto al de los otros. Para empezar, sólo se admitirá un envío al juez. La puntuación del problema no se conocerá hasta el día siguiente, puesto que depende no sólo de tu programa, sino también los demás participantes. Cuando tengas tu programa listo, deberás enviarlo al juez. Sólo puedes hacer un único envío; ignora la respuesta que te de el juez on-line.

Recibirá más puntos quien obtenga más chiqui-puntos superando juegos de prueba particulares, de distintos tamaños y números de barcos. Todos los programas deberán superar los mismos juegos de prueba. Se concederán chiqui-puntos por cada juego de pruebas superado. Un programa que, durante la ejecución de un juego de pruebas, ataca una casilla no existente o escribe `FIN` antes de tiempo no recibirá ningún chiqui-punto para el juego de pruebas donde falla. El programa que tarde más turnos en hundir los barcos de un juego de pruebas recibirá un chiqui-punto; el siguiente, dos; y así hasta el más eficiente. En caso de empate, todos los programas empatados recibirán el máximo número de chiqui-puntos que podrían recibir. Por ejemplo, si 3 programas empatan en la última posición, los 3 recibirán 3 chiqui-puntos; el siguiente programa recibirá 4 chiqui-puntos (si no está empatado), etc.

Al final, se distribuirán los puntos del problema del siguiente modo,

$$\text{puntos} = \left\lceil 100 \cdot \left( 1 - \frac{i-1}{N} \right) \right\rceil,$$

donde  $N$  es el número de soluciones que se reciban y  $i$  es la posición que ocupa en la clasificación acumulada de chiqui-puntos; es decir, el primer clasificado recibirá 100 puntos, y los restantes clasificados recibirán puntos proporcionalmente a su posición. En caso de empate, se aplicará el mismo criterio que antes para repartir los puntos. La clasificación y la puntuación se harán públicas al día siguiente.

**Autor:** Javi Gómez

### Información del problema

Autoría: Omer Giménez

Generación: 2026-01-25T11:10:59.088Z

© Jutge.org, 2006–2026.

<https://jutge.org>