

MODALIDAD B. INDIVIDUAL

IDENTIFICADOR DEL ALUMNO:

CATEGORÍA: JUVENIL (BACH/CF) CADETE (ESO)

HORA COMIENZO PRUEBA:

HORA FINALIZACIÓN PRUEBA:

INSTRUCCIONES DE LA PRUEBA

Esta prueba consta de cuatro (4) ejercicios, de dificultad variada.

En el escritorio encontrarás un directorio llamado OIIAsturias2025. Dentro hay dos directorios.

Uno se llama Participante. Deberás **renombrar** este directorio con tu identificador de participante. Todo el código que desarrolles deberá estar aquí.

En el otro se llama Software Olimpiada, contiene los tres entornos de trabajo que se proporcionan.

El enunciado completo **deberá ser entregado antes de abandonar el aula**.

Cuando termines, debes notificárselo a los profesores encargados de la prueba, que anotarán la hora en que ha sido superada esta prueba.

Cada estudiante dispone de **dos horas y media** para solucionar, de forma individual, estos retos.

En los días siguientes, los profesores valorarán las soluciones de las distintas sedes y otorgarán distinta puntuación a los alumnos que la hayan superado, ateniéndose a los siguientes criterios:

- La corrección de la solución.** Aquellas soluciones que funcionen correctamente o estén muy cerca de hacerlo, serán mejor puntuadas.
- Los ejercicios solucionados.** Los ejercicios tienen distinta dificultad (indicada en el enunciado). Aquellos ejercicios más difíciles, lógicamente, tendrán mayor puntuación si logran resolverlos.
- La claridad de la solución.** La utilización de nombres representativos, comentarios adecuados, procedimientos auxiliares, etc. mejora la calidad de la solución.
- La eficiencia de la solución.** Un algoritmo demasiado complejo para el problema o escribir código para el que existe una función, método, procedimiento, etc., estándar en el lenguaje elegido, restará calidad a la solución.

En caso de empate a puntos, se utilizará el tiempo para desempatar.

No debes preocuparte por asegurarte de que el valor de entrada sea correcto. Puede suponer que siempre cumple las condiciones de cada ejercicio.

En los ejemplos que ilustran los ejercicios, por razones de espacio, pueden aparecer varios números en la misma fila, lo que no significa que deban leerse del teclado en una misma línea. Puede elegir la solución que prefiera.

Se proporciona código auxiliar para leer la entrada. Eres libre de usarlo o no, utilizar una parte, cambiarlo, etc.

EJERCICIO 1: ¡ NO ME SIENTO LAS PIERNAS! (10 PUNTOS)

Mis amigos y yo hemos decidido ir al cine este fin de semana. Lamentablemente, es un cine antiguo y hay muy poco sitio entre butacas para poder estirar las piernas. Así que, dado que nos vamos a pasar allí varias horas, hemos decidido estudiar la sala concienzudamente para elegir las mejores butacas.

Después de mucho discutir, hemos decidido que los mejores asientos son los de la primera fila o los que dan a un pasillo.

Sabemos que el cine numera sus asientos tal como se indica a continuación

PANTALLA								
PASILLO	Butaca 1	Butaca 2	...	...	n-2	n-1	n	PASILLO
	2n	...	...	...	...	n+2	n+1	
	2n + 1	2n + 2	...	...	...	...	3n	
	...	...	...	...	...	...	...	

Como ves, todas las filas tienen el mismo número de butacas.

Por favor, ayúdanos, a elegir los mejores asientos. Escribe un programa que, dado un número de butaca nos diga si es un buen sitio o no.

Entrada

La entrada se compondrá de una línea en la que habrá tres números enteros, todos mayores que 0 (no es necesario comprobarlo). El primero indica el número de filas de la sala, el segundo el número de butacas por fila y el tercero, el número de la butaca que queremos reservar.

Salida

La salida mostrará el mensaje **PRIMERA FILA**, si el asiento está en primera fila, **PASILLO**, si está al lado de un pasillo, o **SIGUE BUSCANDO**, si no está ni en la primera fila ni en el pasillo.

A continuación, se muestran algunos **ejemplos**

EJEMPLO 1	EJEMPLO 2	EJEMPLO 3
Entrada: 3 3 1	Entrada: 3 4 5	Entrada: 3 3 5
Salida: PRIMERA FILA	Salida: PASILLO	Salida: SIGUE BUSCANDO

EJERCICIO 2: INTRUSO (20 PUNTOS)

Para celebrar mi cumpleaños he decidido hacer una fiesta con aforo limitado. No todos pueden acceder, depende de cuántos caben en mi casa.

Como no me veo capaz de elegir yo a mis mejores amigos, quiero una aplicación que asigne un número a cada amigo que se conecte. Al primero que se conecte le asignará el número uno y a todos los demás les asignará el número entero inmediatamente siguiente al que se conectó justo antes.

El día de la fiesta, les he pedido que formen una fila, por orden según el número asignado, pero como conozco a mis amigos y no me fío de ellos, salgo a controlar que no se haya colado nadie. Empezando por el primero de la cola, les pido que me enseñen su turno. Algunos, hartos de esperar se han ido y otros no han mantenido el orden en la cola, y les impido entrar a mi fiesta.

Escribe un programa que reciba los turnos de los que están esperando entrar en mi fiesta, en el orden en que están en la cola y que imprima el turno del primero y del último en entrar.

Entrada

La entrada se compondrá de dos líneas. En la primera línea habrá dos números enteros que indican cuántos amigos pueden acceder a la fiesta (aforo) y cuántos hay en la cola (esperando). El valor de aforo será un número entre 1 y 50 y el de esperando será un número entre 1 y 100. No es necesario comprobar ninguna de las dos condiciones.

En la segunda estarán los turnos que tienen cada uno de esos amigos, según están en la cola. Siempre habrá tantos números como se haya indicado en esperando y serán números entre 1 y 500, todos distintos (tampoco hace falta comprobar ninguna de estas condiciones).

Impediré la entrada a todos aquellos que no estén en el lugar de la cola que les corresponda, según su turno.

Salida

El programa escribirá el turno que tienen el primer y el último invitado en entrar.

A continuación, se muestran varios **ejemplos**.

EJEMPLO 1	EJEMPLO 2	EJEMPLO 3
Entrada: 12 5 2 3 6 7 9	Entrada: 5 8 2 3 6 7 9 12 13 14	Entrada: 10 7 1 4 3 2 5 7 6
Salida: 2 9	Salida: 2 9	Salida: 1 6
Salida: 2 3 6 7 9	Salida: 2 3 6 7 9	Salida: 1 2 5 6

EJERCICIO 3: BATALLA NAVAL. COLOCAR LOS BARCOS (35 PUNTOS)

¿Conoces el juego de los barcos? Tiene muchos nombres: hundir la flota, batalla naval, ...

Dos jugadores juegan con **dos tableros**. Los tableros están divididos en $N \times N$ casillas identificadas con un número para las columnas (del 1 al N) y con una letra para las filas (de la A en adelante).

En uno de los tableros, colocan sus barcos (flota) ocupando una o varias casillas cada uno y van marcando los ataques del rival.

En el otro registran sus disparos a la flota del contrario, diferenciando los impactos de los fallos (agua).

Cada jugador coloca su flota en el primer tablero, en secreto: un **portaaviones** (3 casillas), dos **destruidores** (2 casillas) y tres **submarinos** (1 casilla). Los barcos deben ocupar casillas contiguas en horizontal o vertical. Y no está permitido que dos barcos se toquen en ningún punto.

Ayúdame a comprobar si un jugador ha colocado su flota siguiendo estas normas.

Escribe un programa que verifique si un tablero cumple las reglas: que la cantidad de barcos y su longitud sea correcta y que no se toquen.

Entrada

La entrada comienza con un entero N (entre 5 y 10), que indica el tamaño del lado del tablero. Luego, siguen N filas de N caracteres separados por espacios donde cada carácter puede ser: **0** (agua o casilla vacía), **p** (parte del portaaviones), **d** (parte de un destructor) y **s** (submarino). No es necesario validar la entrada.

Salida

La salida será **"INCORRECTO"** si hay un número incorrecto de barcos, si alguno tiene una longitud errónea o si hay barcos tocándose. De lo contrario, se imprimirá **"CORRECTO"**.

A continuación, se muestran varios **ejemplos**.

EJEMPLO 1	EJEMPLO 2	EJEMPLO 3
Entrada: 5 p p p p 0 0 0 0 0 0 d d 0 s 0 0 0 0 0 0 s 0 0 d d	Entrada: 5 p p p 0 s 0 0 0 0 0 d d 0 s 0 0 0 0 0 0 s 0 0 d d	Entrada: 5 p p p 0 0 0 0 0 0 d s 0 0 0 d 0 s 0 0 0 s 0 0 d d
Salida: INCORRECTO (falta un submarino, el portaaviones ocupa más casillas de las que debe)	Salida: CORRECTO	Salida: INCORRECTO (los submarinos se tocan)

EJERCICIO 4: BATALLA NAVAL. FLOTA HUNDIDA ¿ O NO? (DIFICULTAD 35)

Suponiendo la misma descripción del juego anterior, una vez colocados los barcos, los jugadores juegan por turnos. Cada jugador dispara a la flota de su oponente indicando las coordenadas de una casilla, y registra el resultado del disparo en el segundo tablero. Si esa posición está ocupada por parte de un barco contrario, se señala con una **t** (*¡Tocado!*) si todavía quedan partes del barco a flote, o con una **h** (*¡Hundido!*) si con ese disparo la nave ha quedado totalmente destruida. Si la posición indicada no corresponde a una parte de barco alguno, se marcará con una **a** (*¡Agua!*).

En este ejercicio, debes suponer que el tablero de entrada al juego está correctamente formado. Es decir, el número de barcos es correcto, los barcos tienen la longitud adecuada y no hay barcos tocándose.

Sin embargo, puede suceder que, al marcar los disparos que he realizado, haya anotado mal alguno.

Ayúdame a comprobar si he ganado, si quedan todavía barcos de mi oponente a flote o si me he equivocado al anotar algún disparo. Escribe un programa que lo compruebe.

verifique si todos los barcos de mi oponente están hundidos o no.

Entrada

La entrada comienza con un entero **N** (entre 5 y 10), que indica el tamaño del tablero. Luego, siguen **N** filas de **N** caracteres separados por espacios: **0** (casilla no disparada), **t** (barco tocado), **h** (barco hundido) o **a** (disparo fallido, agua). Estas filas representan un tablero con el número correcto de barcos, la longitud correcta y correctamente colocados. No es necesario validar la entrada.

Salida

La salida será **"GANADOR"** si todos los barcos del oponente están hundidos, **"INCORRECTO"**, si el estado del tablero es incorrecto o **"SIGUE JUGANDO"**, si hay barcos a flote.

A continuación, se muestran varios **ejemplos**.

EJEMPLO 1	EJEMPLO 2	EJEMPLO 3
Entrada: 5 t t 0 0 0 0 0 0 a h h h 0 h 0 0 0 0 a a h 0 0 a t	Entrada: 5 h h h 0 0 0 0 0 0 0 h h 0 h 0 0 0 a 0 0 0 0 0 0 0	Entrada: 5 h h h 0 0 0 0 0 0 h h 0 h 0 h 0 0 0 0 0 h 0 0 h h
Salida: INCORRECTO (hay un barco tocado que tiene agua 'a' alrededor. O bien debería estar hundido o una de las 'a' está mal colocada)	Salida: SIGUE JUGANDO	Salida: GANADOR