EJERCICIO 1 – Tienda de objetos mágicos

En una tienda de un mundo de fantasía hay 6 objetos mágicos, cada uno con su precio. Se almacenan en **dos arrays** paralelos:

- **Objetos:** poción, antorcha, daga, escudo, arco, grimorio.
- **Precios:** 10.0, 5.0, 25.0, 40.0, 60.0, 120.0.

El jugador empieza con una cantidad de oro que el usuario introducirá. El programa debe:

- Mostrar todos los objetos con sus precios usando un método llamado **mostrarTienda** que reciba los dos arrays como parámetros.
- Pedir al usuario que introduzca el nombre de un objeto para comprar, mientras le quede suficiente oro.
- Crear las funciones:
    - **mostrarTienda**: muestra los objetos disponibles en la tienda junto con sus precios.
    - **buscarObjeto**: se le pasa un objeto como argumento y lo busca en el array de objetos. En caso de encontrarlo, devuelve su índice, y en caso de que no, devuelve -1.
    - **puedeComprar**: esta función recibe como argumento el oro que le queda al jugador y el precio de un objeto. Devuelve un booleano que nos dice si el usuario puede o no comprar ese objeto con el oro que le queda.
- Funcionamiento del programa:
    - Si el objeto que ha introducido el usuario no existe en la tienda se debe mostrar mensaje de error.
    - Si el objeto que ha introducido el usuario sí existe, pero el oro no le alcanza para poder comprarlo, se debe informar de ello y terminar el proceso de compra.
    - En el caso de que el objeto sí exista y el jugador tiene oro suficiente para comprarlo, se debe restar el coste al oro del jugador. Por último, mostrar el oro restante y permitir seguir comprando.
- Cuando el jugador se quede sin oro suficiente para comprar ningún producto, se debe finalizar el programa y mostrar el oro que le queda.

EJERCICIO 2 – Mochila del aventurero

La mochila del jugador tiene una **capacidad máxima de peso** (que el usuario introduce). El jugador quiere guardar objetos hasta llenar la mochila.

Se dispone de un array con **pesos de objetos encontrados**:

- **pesos** = [2.0, 1.5, 3.0, 5.0, 0.5, 4.0, 2.5]

El programa debe:

- Pedir al usuario que introduzca el peso de cada objeto que quiere intentar guardar, pero solo puede usar los valores del array anterior.

- Métodos a crear:

- **existePeso**: hace una búsqueda lineal dentro del array de pesos para comprobar si el peso que se ha pasado como argumento es válido. Esta función devuelve un booleano que indica si el peso existe o no.
- **cabeEnLaMochila**: este método recibe el peso actual de la mochila, la capacidad total y el objeto que se quiere meter. Devuelve un booleano que indica si entra o no el objeto en la mochila.

- El programa continuará añadiendo objetos mientras haya espacio.
- Cuando ya no entran más objetos en la mochila se termina el programa, al final debe mostrar:
  - El **peso total final**
  - El **número de objetos guardados**

EJERCICIO 3 – Mapa del Bosque Encantado

El aventurero se mueve por un **mapa 2D del bosque**, representado por una matriz de 5 filas × 6 columnas:

- 0 = zona transitable
- 1 = obstáculo (rocas, árboles gigantes)
- 2 = el tesoro

0	1	0	0	0	1
0	0	0	1	0	0
1	1	0	0	0	0
0	0	0	1	2	1
0	1	0	0	0	0

El programa debe:

1. Mostrar el mapa por pantalla con un método llamado **mostrarMapa**.
2. Pedir al usuario una **coordenada (fila y columna)** donde quiere que el aventurero se mueva.
3. Validar que:
   - Está dentro del rango. Desde una casilla un jugador solamente puede moverse a las casillas de alrededor, por ejemplo, en el caso del siguiente mapa, donde la casilla azul es en la que se encuentra el jugador, las casillas naranjas son aquellas que están dentro de rango.

0	1	0	0	0	1
0	0	0	1	0	0
1	1	0	0	0	0
0	0	0	1	2	1
0	1	0	0	0	0

- No es un obstáculo (1).

4. Métodos obligatorios:
    - **esTransitable:** este método recibe una fila y una columna, y devuelve un booleano que indica si la casilla es transitable o no (es o no un obstáculo).
    - **esTesoro**: este método recibe las coordenadas de una casilla e indica si es o no el tesoro.
5. Funcionamiento:
    - Si la casilla a la que se quiere mover el jugador está dentro de rango y es transitable, mostrar “Puedes avanzar”.
    - Si la casilla a la que se quiere mover el jugador es el tesoro, mostrar “¡Has encontrado el tesoro!”
    - Si la casilla a la que se quiere mover el jugador es un obstáculo, mostrar “No puedes pasar por ahí”.