Crea un **paquete** llamado **primeraevaluacion**. Realiza una clase por cada ejercicio llamada EjercicioX donde X sea el número del ejercicio. La puntuación aparece junto a cada ejercicio. Se valorará no sólo que los ejercicios sean correctos, hagan lo que se dice que tienen que hacer, si no que estén bien ***tabulados***, haya ***comentarios*** en las funciones y donde haya algo que aclarar, que crees ***métodos*** cuando sea necesario, que se controlen las ***excepciones***, etc.

1. [2 puntos] Nos piden crear una matriz 20×20 de números enteros que inicialmente rellenamos de valores aleatorios (1-100), nos piden hacer un **menú** con estas opciones:

    1. Intercambiar columna. Te pedirá dos números de columna e intercambiará los valores de la primera por los valores de la segunda.
    2. Suma de una fila que se pedirá al usuario (controlar que elija una correcta)
    3. Comprueba si la diagonal principal y la inversa son iguales.
    4. Pintar las coordenadas i,j donde se encuentra el menor elemento de la matriz.
    5. Muestra la matriz con las filas ordenadas de menor a mayor
    6. Salir

2. [3 puntos] **El Reino de los Dados Mágicos**
   Dos jugadores se enfrentan en un duelo usando *dados mágicos* con poderes especiales. Cada ronda introduce una regla especial que puede cambiar la puntuación o las condiciones del duelo.

   Preparación inicial:
   - Ambos jugadores comienzan con 40 puntos de vida.
   - Se decide el número de rondas (mínimo 5).
   - Los jugadores usan dados de 6 caras.

   Desarrollo de cada ronda:
   - Antes de lanzar los dados, se selecciona una regla mágica de forma aleatoria.
   - Ambos jugadores lanzan su dado, y el resultado se interpreta según la regla mágica de esa ronda.
   - Según el resultado, los jugadores pueden ganar o perder puntos de vida.

   Reglas mágicas posibles (se elige una al azar en cada ronda):
   - ***Dado de Fuego***: El perdedor (menor resultado) de la ronda pierde tantos puntos de vida como la diferencia entre ambos resultados.
   - ***Dado de Curación***: Ambos jugadores suman el resultado de su dado a su vida.
   - ***Dado de Robo***: El jugador con el dado más alto roba puntos de vida al otro igual a la mitad de su resultado (redondeado hacia abajo).
   - ***Dado Explosivo***: Si un jugador saca un 6, el otro pierde automáticamente 10 puntos de vida. Si ambos sacan 6, se anula el efecto.
   - ***Dado de Escudo***: En esta ronda, ningún jugador pierde puntos de vida, pero el ganador (mayor puntuación) recibe un "escudo" que le protege de daño en la siguiente ronda.

Final del juego:

- El duelo termina si uno de los jugadores llega a 0 puntos de vida. El otro será el ganador.
- Si se completan todas las rondas y ambos siguen vivos, gana el jugador con más puntos de vida.

3. [2 puntos] **Codificador de mensajes**

Crea un **método** en Java que tome una frase como parámetro y aplique varias transformaciones para codificar un mensaje de salida según las siguientes reglas:

a. Primera letra al final: Mueve la primera letra de cada palabra al final de la palabra.
b. Reemplazo de vocales: Reemplaza las vocales de la palabra con caracteres especiales: a -> @, e -> &, i -> $, o -> º, u -> #.
c. Invertir cada palabra: Si la longitud de la palabra es impar, invierte la palabra resultante antes de añadirla al mensaje codificado.

Realiza otro método para desencriptar el mensaje y comprueba que funciona desencriptando el mismo mensaje encriptado, y viendo que el resultado es el original.

4. [3 puntos] Escribe un programa en Java que simule un juego de la **Caza del Tesoro**. En este juego, el tablero estará representado por una matriz 10x10 y el objetivo es encontrar el tesoro escondido en una de las celdas del tablero. El jugador debe ingresar coordenadas para intentar encontrar el tesoro.

Reglas del juego:

a. El tablero tiene dimensiones de 10x10, inicialmente vacías '-'.
b. Se coloca un tesoro en una casilla aleatoria del tablero.
c. El jugador hará intentos para localizar el tesoro, dando la coordenada x y la coordenada y donde cree que está el tesoro.
d. El jugador tiene **15 intentos** para encontrar el tesoro.
e. Después de cada intento, el juego dará pistas:
   - Si el jugador dispara a una casilla más cerca del tesoro (en términos de distancia), se le da la pista "*Estás más cerca del tesoro*".
   - Si el jugador dispara a una casilla más alejada, se le da la pista "*Estás más lejos del tesoro*".
   - La distancia puede calcularse usando la **distancia de Manhattan** que se calcula como la suma del **valor absoluto** de la resta de las coordenadas x, más el valor absoluto de la resta de las coordenadas y. Donde (x1,y1) es el punto donde está el tesoro, y (x2,y2) es el punto donde tú pruebas.

$$\textbf{Distancia} = |x1 - x2| + |y1 - y2|$$

f. El programa debe mostrar el estado del tablero después de cada disparo, con las casillas marcadas como "descubiertas" o "fallos".
g. El jugador debe seguir intentando hasta encontrar el tesoro o agotar sus intentos.
h. Si el jugador encuentra el tesoro, el juego termina y se muestra un mensaje de victoria. Si se quedan sin intentos, se muestra un mensaje de derrota.