

1. Dado un número entero n , diseñar un método

```
public static int transformar(int n)
```

que permita transformar el número original n en otro en el que aparezcan las cifras ordenadas en secuencia ascendente y sin ceros a la izquierda. Ej 85401053 → 134558 (sin los dos ceros). Ayuda: utilizar un array para almacenar las cifras del número.

2 puntos

2. Dado el array de caracteres `nif` de contenido:

`T R W A G M Y F P D X B N J Z S Q V H L C K E`

y el array `dni` de tamaño `TAM` y n elementos (`int [] dni, int n`),

diseñar un método que reciba el `dni` y nos permita obtener la letra CIF que le corresponde, sabiendo que la letra es la que ocupa la posición en el array `nif` igual al resto de dividir el `dni` por 23.

Por ejemplo para un `dni` 3 4 7 9 2 7 3 representa el número 3479273

$3479273 \% 23 = 17$ si T ocupa la pos 0, la H ocupa la pos 17, el método devolvería la letra H

2 puntos

3. Diseñar un método `public static int paresImpares(int [] a, int n)` que recibe un array `a` de tamaño `TAM` de n números, ordenar el array colocando en la parte superior todos los pares y en la parte inferior los impares. No se podrá crear ningún otro array y se ordenan los pares y los impares por separado. El método devolverá el número de pares contenido en el array.

Paso 1, separo pares e impares 8 4 22 2 6 34 54 62 | 7 3 5 21 33 1

Paso 2, ordeno 2 4 6 8 22 34 54 62 | 1 3 5 7 21 33

3 puntos

4. Dado un array `a` de n números enteros y tamaño `TAM`, diseñar un método que calcule el tamaño de la secuencia más larga de elementos repetidos ocupando posiciones consecutivas y el valor de este elemento repetido. `public static int [] repetidos (int [] a, int n)`

El método devolverá un array de dos posiciones, en la primera el tamaño de la secuencia más repetida y en la segunda el valor repetido.

a: 3,5,5,4,3,6,6,6,6,1,1,1,2,2,2,2,2,4,4 sería 6 (veces repetido el 2)

la respuesta `res[0]` 6 y `res[1]` 2

3 puntos