Programación

Control 2ª Evaluación - Colecciones

Problema 1. (3 ptos.)

Realiza un programa Java que dada una cadena de caracteres introducida por teclado, cuya longitud es arbitraria, muestre como resultado la lista de caracteres que contiene, junto el nº de veces que cada uno de ellos aparece en la cadena, ordenada ascendentemente por el nº de ocurrencias.

Además, mostrará el carácter que más veces aparece en la cadena y ese nº de veces.

Para resolver el problema, utiliza el API de las colecciones Java.

Ejemplo de ejecución:

```
Introduce frase: Hola me llamo Lola
{a=3, e=1, H=1, l=4, L=1, m=2, o=3}

El caracter más frecuentes es: l
Aparece: 4 veces
```

Problema 2. (2,5 ptos.)

Realiza un programa Java que genere una matriz como la siguiente:

	1	2	3	4	5	6	7	8	9	10
1	1	2	3	4	5	6	7	8	9	10
2	2	4	6	8	10	12	14	16	18	20
3	3	6	9	12	15	18	21	24	27	30
4	4	8	12	16	20	24	28	32	36	40
5	5	10	15	20	25	30	35	40	45	50
6	6	12	18	24	30	36	42	48	54	60
7	7	14	21	28	35	42	49	56	63	70
8	8	16	24	32	40	48	56	64	72	80
9	9	18	27	36	45	54	63	72	81	90
10	10	20	30	40	50	60	70	80	90	100

* *Los números de filas y columnas (en rojo) son aclaratorios, no se deben de almacenar.*

Problema 3. (2 ptos)

Realiza un programa Java que, a partir de un array de enteros inicialmente desordenado, determine la **mediana**. La mediana es el valor que ocupa la posición central del array una vez ORDENADO ascendentemente. En caso de existir dos elementos centrales, la mediana es el promedio de estos. Ejemplo:

5	12	15	**30**	37	40	41

Mediana = **30**

1	7	16	**18**	**25**	33	34	38

Mediana = (18+25)/2 = **21.5**

NOTAS:

- La creación e inicialización del array se hará directamente en código, usando los datos del ejemplo.
- Proporciona 2 formas de ordenar el array.

Ejemplo de ejecución:

```
Array: 5 | 12 | 15 | 30 | 37 | 40 | 41
   Mediana=30
Array: 1 | 7 | 16 | 18 | 25 | 33 | 34 | 38
   Mediana=21,5
```

Problema 4. (2,5 ptos)

Diseña la clase *Persona* teniendo en cuenta las siguientes indicaciones:

- Atributos:
  - idPersona: identificador único de la persona. Numérico y correlativo.
  - nombre: cadena con el nombre de la persona.
  - sexo: carácter que indica el sexo de la persona. Valores: H (hombre) o M (mujer).
  - padre: de tipo *Persona*.
  - madre: de tipo *Persona*.
- Constructor/es:

  Se podrá crear una persona proporcionando su nombre, o además del nombre, proporcionando su padre y su madre. En este caso, si alguno de ellos (padre o madre) se omite se lanzará una excepción.

- Métodos:
  - toString(): devuelve la información de la persona con el siguiente formato:

```
Persona (ID: <idPersona> Nombre: <nombre> Sexo: <sexo>
Padre: <nombre padre> Madre: <nombre madre>
```

  - sonHermanos(Persona otra): determina si la persona actual y otra persona recibida como parámetro son hermanos. Dos personas son hermanos si tienen el mismo padre y la misma madre.
  - huerfano(): método que deja a la persona actual sin padre y sin madre.
  - familiaHomoparental(): método que determina si la familia de la persona está formada por progenitores del mismo sexo, es decir, dos padres o dos madres.