

Completa los métodos de la clase *Utilidades*. Los requerimientos de funcionamiento están descritos en el comentario predecesor de cada uno de ellos.

```
public class Utilidades
{
    private Utilidades()
    {
    }

    /**
     * Devuelve un array nuevo con los mismos números del array nums de entrada
     * pero sin los números impares que pudiera contener.
     */
    public static int[] eliminaImpares(int[] nums)
    {
        . . .
    }

    /**
     * Devuelve una cadena nueva con los mismos caracteres y en el mismo
     * orden que la cadena de entrada, pero eliminando los caracteres
     * numéricos que pudiera contener.
     */
    public static String eliminaNumeros(String cadena)
    {
        . . .
    }

    /**
     * Devuelve un array nuevo resultante de la unión de los números
     * de los dos arrays de entrada.
     */
    public static int[] union(int[] nums1, int[] nums2)
    {
        . . .
    }
}
```

Ejemplo de ejecución:

```
public class Main
{
    public static void main(String[] args)
    {
        int[] nums1 = {0,1,2,3,4,5};
        int[] nums2 = {6,7,8};
        int[] pares = Utilidades.eliminaImpares(nums1);
        muestraArray(pares);
        System.out.println(Utilidades.eliminaNumeros("213abc0AB99C7 fin"));
        int[] union = Utilidades.union(nums1, nums2);
        muestraArray(union);
    }

    private static void muestraArray(int[] nums)
    {
        for (int i = 0; i < nums.length; i++)
        {
            System.out.print(nums[i] + " ");
        }
        System.out.println();
    }
}
```

Salida por consola:

```
0 2 4
abcABC fin
0 1 2 3 4 5 6 7 8
```

Completa la codificación de la clase Ciudad, encargada de gestionar datos referentes a sus temperaturas registradas, teniendo en cuenta que en una ciudad se pueden registrar un número indeterminado de temperaturas.

```
public class Ciudad
{
    // Atributos de clase:
    . . .

    // Constructor:
    public Ciudad(String nombre)
    {
        . . .
    }

    // Devuelve el nombre de la ciudad:
    public String getNombre()
    {
        . . .
    }

    // Registra una temperatura:
    public void registrarTemperatura(float temperatura)
    {
        . . .
    }

    // Devuelve la temperatura máxima de las registradas:
    public float getTemperaturaMaxima()
    {
        . . .
    }

    // Devuelve la temperatura mínima de las registradas:
    public float getTemperaturaMinima()
    {
        . . .
    }

    // Devuelve la temperatura media de las registradas:
    public float getTemperaturaMedia()
    {
        . . .
    }
}
```

Sea la clase Main, donde el método *cargarCiudades* se encarga de crear un array de ciudades, cada una de ellas con su nombre y varias temperaturas registradas (al menos una).

```
public class Main
{
    public static void main(String[] args)
    {
        Ciudad[] ciudades = cargarCiudades();
        . . .
    }
}
```

Completa el código para mostrar un listado de datos sobre las ciudades cargadas y finalmente el nombre de la ciudad más fría con el formato de salida propuesto.

Ejemplo de ejecución:

```
Madrid: Máxima: 43.762474 Mínima: -9.800246 Media: 18.916195
Barcelona: Máxima: 43.000557 Mínima: -6.6271276 Media: 20.077631
Valencia: Máxima: 41.521603 Mínima: -4.644973 Media: 12.112326
Albacete: Máxima: 37.98532 Mínima: -6.745476 Media: 11.231459
Cuenca: Máxima: 42.93141 Mínima: -9.931812 Media: 11.142011
Zaragoza: Máxima: 38.38445 Mínima: 0.4395485 Media: 20.98256
```

```
Más fría de media: Cuenca
```