

Escribir una clase con un método **obtenerArrCad5VocalesAL**, que reciba por parámetro un array de cadenas y devuelva un array con las que contengan las 5 vocales. Deberá utilizarse como elemento de almacenamiento temporal un objeto de la clase `ArrayList<String>`. Para la consideración de un carácter como vocal no se tendrá en cuenta si está en mayúsculas o en minúsculas.

El método main que probará el funcionamiento del método anterior será:

```
public static void main(String[] args) {
    String[] cadenas={null, "aEiou", "ccc","Aeina","aceituno"};
    // String[] cadenas=null;
    /*Probad el funcionamiento con las 2 definiciones
    anteriores*/
    try{
        String[] c5v=obtenerArrCad5Vocales(cadenas);
        System.out.println(Arrays.toString(c5v));
    }
    catch(Exception e){
        System.out.println(e.getMessage());
    }
}
```

Planteamiento:

(Este planteamiento es el que deberías hacer ante un enunciado como el anterior antes de comenzar a codificar el programa)

1. Se comprueba que el array que se recibe como parámetro no es una referencia nula (es distinto de null). En caso de ser null, se lanzará una excepción de tipo ***IllegalArgumentException*** que muestre el mensaje "Parámetro no válido"
 2. Se crea una lista (un objeto de la clase `ArrayList<String>`) que almacenará las cadenas con 5 vocales.
 3. Se recorre el array que se recibe como parámetro y para cada elemento se comprueba si cumple la condición (si tiene las 5 vocales) y si la cumple se añade a la lista
 4. Finalmente se crea un array, con el contenido de la lista, que será el que se devuelva como resultado.
- Intentad utilizar en cada punto los métodos de las clases `String`, `Array` o `ArrayList` que faciliten la tarea.
 - El método debe continuar su funcionamiento y aunque el tratamiento de alguno de los elementos del array original pueda provocar un error (Ej. El null)