

1. Método que recibe un ArrayList de enteros y devuelve una matriz irregular de enteros con valores aleatorios entre 1 y 200. Cada elemento del ArrayList contiene la longitud de cada fila de la matriz.

Escribe un método que reciba la matriz obtenida en el método anterior y la muestre por pantalla

```
public static int[][] metodo(ArrayList<Integer> A) {
    Random rnd = new Random();
    int[][] matriz = new int[A.size()][];
    for (int i = 0; i < matriz.length; i++) {
        matriz[i] = new int[A.get(i)];
    }
    for (int i = 0; i < matriz.length; i++) {
        for (int j = 0; j < matriz[i].length; j++) {
            matriz[i][j] = rnd.nextInt(200) + 1;
        }
    }
    return matriz;
}

public static void mostrar(int [][] A){
    for(int i = 0; i < A.length; i++){
        for (int j = 0; j < A[i].length; j++) {
            System.out.printf("%5d", A[i][j]);
        }
        System.out.println();
    }
}
```

2. Escribe un **método** que reciba un String que contiene NIFs y devuelva un String con los NIF ordenados de menor a mayor. Por ejemplo, para el String: "43434321A 65473820T 33238765D 32994102X 74771234F"

El método devolverá el String: "32994102X 33238765D 43434321A 65473820T 74771234F"

```
public static String metodo(String cadena){
    String [] array = cadena.split(" ");
    Arrays.sort(array);
    StringBuilder sb = new StringBuilder();
    for(String s : array){
        sb.append(s);
        sb.append(" ");
    }
    return sb.toString().trim();
}
```

3. Un palíndromo es una frase que se lee igual de derecha a izquierda que de izquierda a derecha sin tener en cuenta espacios en blanco, signos de puntuación ni acentos. Por ejemplo:

Dábale arroz a la zorra el Abad

Lavan esa base naval

Isaac no ronca así

Escribe un **método** que reciba una cadena de caracteres (String) y compruebe si esa cadena es un palíndromo.

El método devolverá true si se trata de un palíndromo o false si no lo es. Para simplificar el ejercicio supondremos que las palabras están separadas por espacios en blanco y que el texto no contiene signos de puntuación ni vocales acentuadas.

```
public static boolean palindromo(String cadena) {
    String[] array = cadena.split(" ");
    StringBuilder original = new StringBuilder();
    for (String s : array) {
        original.append(s);
    }
    StringBuilder reverso = new StringBuilder(original);
    reverso.reverse();
    return original.toString().equalsIgnoreCase(reverso.toString());
}
```

4. Escribe un **método** que recibe un texto (String) con palabras separadas por espacios en blanco, comas y puntos y devuelve un String con todas las palabras del texto separadas por comas y ordenadas alfabéticamente en orden descendente (de la z a la a). Además si alguna palabra tiene menos de 4 caracteres el String resultante se devolverá entre paréntesis.

Por ejemplo: si el método recibe el texto: "naranja, uva, manzana, pera, lechuga tomate"

Devolverá el String entre paréntesis: "(uva, tomate, pera, naranja, manzana, lechuga)"

Si el método recibe el texto: "naranja, cebolla, manzana, pera, lechuga tomate"

Devolverá el String sin paréntesis: "tomate, pera, naranja, manzana, lechuga, cebolla"

```
public static String metodo(String s) {
    StringTokenizer st = new StringTokenizer(s, " ,.");
    StringBuilder sb = new StringBuilder();
    ArrayList<String> palabras = new ArrayList();
    boolean parentesis = false;
    while (st.hasMoreTokens()) {
        palabras.add(st.nextToken());
    }
    Collections.sort(palabras, Collections.reverseOrder());
    for (int i = 0; i < palabras.size(); i++) {
        sb.append(palabras.get(i));
        if (palabras.get(i).length() < 4) {
            parentesis = true;
        }
        if (i != palabras.size() - 1) {
            sb.append(",");
        }
    }
    if (parentesis) {
        sb.insert(0, "(");
        sb.append(")");
    }
    return sb.toString();
}
```

5. Escribe un **método** que reciba un array estático de enteros y devuelva, una matriz irregular de enteros de dos filas. La primera fila de la matriz contendrá los números pares del array ordenados de menor a mayor y la segunda fila contendrá los números impares del array ordenados de mayor a menor. Consideramos el cero como número par.

Ejemplo: Si el método recibe el array:

6	0	-1	7	14	9	-1	20	32	44
---	---	----	---	----	---	----	----	----	----

Devolverá la matriz:

0	6	14	20	32	44
9	7	-1	-1		

```

public static int[][] metodo(int[] A) {
    int[][] m = new int[2][];
    ArrayList<Integer> pares = new ArrayList();
    ArrayList<Integer> impares = new ArrayList();
    for (int i = 0; i < A.length; i++) {
        if (A[i] % 2 == 0) {
            pares.add(A[i]);
        } else {
            impares.add(A[i]);
        }
    }
    m[0] = new int[pares.size()];
    m[1] = new int[impares.size()];
    Collections.sort(pares);
    Collections.sort(impares, Collections.reverseOrder());
    for (int i = 0; i < pares.size(); i++) {
        m[0][i] = pares.get(i);
    }
    for (int i = 0; i < impares.size(); i++) {
        m[1][i] = impares.get(i);
    }
    return m;
}

```

6. Escribe un **método** que reciba dos ArrayList A y B con elementos de tipo entero. El ArrayList A no contiene valores repetidos. El método modifica los ArrayList de esta forma:

Se obtiene el mayor elemento de A y se suma este valor a todos los elementos del array B.

Se obtiene la suma de los elementos de B y se sustituye el valor del elemento mayor de A por la suma de los elementos de B.

Ejemplo:

```

ArrayList A = [2, 5, -4, 3, 10, -5, -15, -3]
ArrayList B = [8, 4, 3, 10, 1]
Mayor de A = 10
Nuevo contenido del ArrayList B = [18, 14, 13, 20, 11]
Suma de los elementos de B = 76
Se sustituye el mayor de A (10) por la suma de B (76):
ArrayList A = [2, 5, -4, 3, 76, -5, -15, -3]

```

```

public static void metodo(ArrayList<Integer> A, ArrayList<Integer> B) {
    int mayor = A.get(0), suma = 0;
    for (int i = 1; i < A.size(); i++) {
        if (A.get(i) > mayor) {
            mayor = A.get(i);
        }
    }
    for (int i = 0; i < B.size(); i++) {
        B.set(i, B.get(i) + mayor);
        suma += B.get(i);
    }
    A.set(A.indexOf(mayor), suma);
}

```

7. Escribe un **método** que reciba una matriz irregular de String y un carácter y devuelva un ArrayList con todas las palabras de la matriz que contengan el carácter recibido excepto aquellas palabras que lo contengan en la primera posición.

Por ejemplo, si el método recibe la matriz:

patata	pimiento	naranja		
ajo	cebolla			
albahaca	perejil	zanahoria	puerro	aguacate

Y recibe el carácter: 'a '

Devolverá el ArrayList: [patata, naranja, cebolla, zanahoria]

```
public static ArrayList<String> metodo(String [][] A, char c){
    ArrayList<String> palabras = new ArrayList();
    for(int i = 0; i < A.length; i++){
        for(int j = 0; j < A[i].length; j++){
            if(A[i][j].indexOf(c) != -1 && A[i][j].charAt(0) != c){
                palabras.add(A[i][j]);
            }
        }
    }
    return palabras;
}
```