NOMBRE		ETAPA / CICLO	CURSO
		CFGS DAW/DAM	1º
APELLIDOS		ASIGNATURA/MÓDULO	CONVOCATORIA
		PROGRAMACION	CONTINUA
DNI	FECHA	NOTA	
	18/10/2023		

1. **(1p)** Transform the following code (i.e., refactor it) to have an if-else statement instead. The result of both codes must be the same.

```
switch (numero) {
    case 0:
    case 1:
        System.out.println("Mountains");
        break;
    case 2:
        System.out.println("Ocean");
    case 3:
        System.out.println("Sea");
        break;
    case 4:
        System.out.println("Desert");
        break;
    default:
        System.out.println("Nature");
}
```

2. **(1p)** Transform the following code into a "while" loop statement to perform the same action.

```
public static void main(String[] args) {
    for (int i = 10, j = 1; i > 0; i--, j++) {
        System.out.println(i + " " + j);
    }
}
```

3. **(2p)** Create a function that, given a size, creates the following matrix:

Enter a number: 5

```
1 1 1 1 1
0 2 2 2 2
0 0 3 3 3
0 0 0 4 4
0 0 0 0 5
```

4. **(2p)** A junior developer can't compile the following code. Help this person modify anything you think it must be changed (be aware of castings and conversion between Strings and numbers)

```
public static void badCode(){
    int num = 'X' + 1.73;
    System.out.println(num); // the console must display letter Y
    double num2 = "44.6" + 1;
    System.out.println(123 + num2); // the console must display
12345.6
    System.out.println('7' + '-' + '3'); // the console must
display 7-3
    double num3 = 1.5f + 1.5d;
    int num4 = "3" + num3;
    System.out.println(num4);  //the console must display 6
    double num5 = 2.5;
    float num6 = num5 + 2;
    System.out.println(num6); //the console must display 4.5
}
```

5. **(2p)** Create a function that generates a random password of a given size made up of uppercase and lowercase letters and numbers. Create a main method that uses the created function to display a password of length 6.

   Random password of length 6: e6lO9H

   Hint. You can create function that return a String with all allowed characters.

6. **(1p)** Create the function equals(String text1, String text2) that compares text1 with text2. The result is true if and only if the text1 represents the same sequence of characters as text2.

   To solve this exercise, you only can use String methods length() and charAt(int index).

7. **(1p)** Create the function endsWith(String text, String suffix) that tests if this string ends with the specified suffix.

   Parameters:

   a. text – text to check.
   b. suffix - the suffix.

   Returns: true if the character sequence represented by the argument is a suffix of the character sequence represented by this object; false otherwise. Note that the result will be true if the suffix argument is the empty string or is equal to this String object as determined by the equals(Object) method.

   To solve this exercise, you only can use String methods length(), equals() and charAt(int index).