1 Practice 3.1. Control structures and functions

> This practice consists of a series of exercises. Create a folder inside your local server with the name `prac03` and do each exercise in a new file inside the folder.
>
> **Important**: Write all the functions in a file named `functions.php` and call each function in another file, using a file for each exercise.
>
> The inclusion of comments in the scripts will be valued.
>
> Once done, compress the folder, add your name to the compressed file and submit it via **Aules**.

1.1 Exercise 1

Write in the file `functions.php` a function named `checkContactDate` that accepts a string containing a date in the format "yyyy-mm-dd" and returns true if the date is correct. To be correct, the date must accomplish the next requirements:

- The year must be between 1900 and 2100.
- The month must be between 1 and 12.
- The days must be between:
  - 1 and 31 if the month is 1, 3, 5, 7, 8, 10, 12.
  - 1 and 30 if the month is 4, 6, 9, 11.
  - 28 if the month is 2 and the year is not leap year; 29 if the month is 2 and the year is a leap year.

Optionally, you can create a function to calculate if a year is a leap year.

The function must be **strictly typed**, both the input parameters and the output values.

Use the function in the main script to check if various dates are correct. If they are incorrect, show the proper error message.

1.2 Exercise 2

Write in the file `functions.php` a function named `checkBirthday` that accepts a string containing a date in the format "yyyy-mm-dd" and returns the number of days until the birthday. For example, if the input data is "1990-10-10" and today is "2023-10-15", the function must return 5.

You must use **DateTime** objects.

Then, in the main script, if the number of returned days is 0, show the message:

- *Today is your birthday.*

If the number of days is between 1 and 7, show the message:

- *There are X days left for your birthday.*

Use the previous function to check if the input date is valid. If it's not valid, return `null`.

Remember to comment all the functions!!

1.3 Exercise 3

Taking the next array:

```php
$students = array(
    array("id"=> 101, "name"=> "Mike Molina", "grade"=> 5),
    array("id"=> 102, "name"=> "Mery Jane Smith", "grade"=> 8),
    array("id"=> 104, "name"=> "Arthur McFly", "grade"=> 4),
    array("id"=> 112, "name"=> "Lory Grimes", "grade"=> 1),
    array("id"=> 120, "name"=> "Carla Fontana", "grade"=> 6),
    array("id"=> 121, "name"=> "Abdul Bahar", "grade"=> 10)
);
```

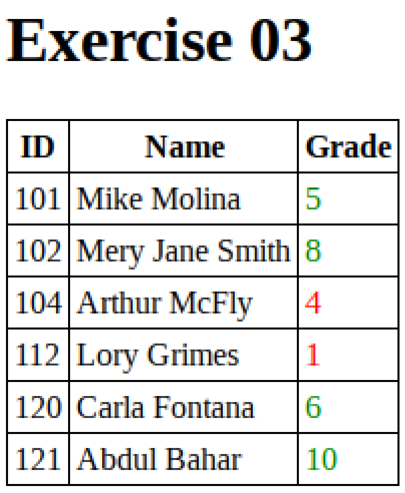
write a function that takes the array as argument and shows all the data formatted as a table. The function must take a second optional argument, an array of strings, for the table header (if no header is provided, use the keys as header). So, the function signature must be:

```
function showStudentsTable (array $data, ?array $header)
```

Use **strict types**.

In the grade column, if the grade is lower than 5, it will be coloured in red, otherwise in green. Use a **`foreach`** loop and format the table in the style you prefer.

Example:



Exercise 03

ID	Name	Grade
101	Mike Molina	5
102	Mery Jane Smith	8
104	Arthur McFly	4
112	Lory Grimes	1
120	Carla Fontana	6
121	Abdul Bahar	10

1.4 Exercise 4

Make a generic version of the previous function, named `showTable`. It should take an associative array of any number of pairs key/value (columns). Don't worry about the grade color. Use **strict types** and the same function signature.