- Submission rules are specified for each task.
- Each task file should properly run and output the correct results.
- code style, code tidiness, comments are a plus.

TASK 1

- **Language**: Javascript

- **Description**:
  Write a function for reversing numbers in binary. For instance, the binary representation of 13 is 1101, and reversing it gives 1011, which corresponds to number 11.

- **How to submit**:
  Complete the source code file named `reverse_binary.js`.

TASK 2

- **Language**: PHP

- **Description**:
  Write a function that provides change directory (cd) function for an abstract file system.
  Notes:
    - root path is '/'.
    - path separator is '/'.
    - parent directory is addressable as '..'.
    - directory names consist only of English alphabet letters (A-Z and a-z).
    - the function will not be passed any invalid paths.
    - do not use built-in path-related functions.

  For example:
  $path = new Path('/a/b/c/d');
  $path->cd('../x');
  echo $path->currentPath;
  should display '/a/b/c/x'.

- **How To Submit**:
  Complete the source file named `change_directory.php`.

TASK 3

- **Language**: Python

- **Description**:
  Suppose you have:
  - a `haversine(lat1, lng1, lat2, lng2)` function that returns the distance (measured in km) between the coordinates of two given geographic point (`lat` and `lng` are latitude and longitude)
  - an array of geographical zones (`locations`):

    ```
    locations = [
      {'id': 1000, 'zip_code': '37069', 'lat': 45.35, 'lng': 10.84},
      {'id': 1001, 'zip_code': '37121', 'lat': 45.44, 'lng': 10.99},
      {'id': 1001, 'zip_code': '37129', 'lat': 45.44, 'lng': 11.00},
      {'id': 1001, 'zip_code': '37133', 'lat': 45.43, 'lng': 11.02},
      ...
    ];
    ```
  - an array of shoppers:

    ```
    shoppers = [
        {'id': 'S1', 'lat': 45.46, 'lng': 11.03, 'enabled': true},
        {'id': 'S2', 'lat': 45.46, 'lng': 10.12, 'enabled': true},
        {'id': 'S3', 'lat': 45.34, 'lng': 10.81, 'enabled': true},
        {'id': 'S4', 'lat': 45.76, 'lng': 10.57, 'enabled': true},
        {'id': 'S5', 'lat': 45.34, 'lng': 10.63, 'enabled': true},
        {'id': 'S6', 'lat': 45.42, 'lng': 10.81, 'enabled': true},
        {'id': 'S7', 'lat': 45.34, 'lng': 10.94, 'enabled': true},
    ];
    ```

  The goal is to calculate the percentage of the zone covered by enabled shoppers (`coverage`). One shopper covers a zone if the distance among the coordinates is less than 10 km.
  Resulted array should be sorted (desc) as the following one:

  ```
  sorted = [
    {shopper_id': 'S3', 'coverage': 72},
    {shopper_id': 'S1', 'coverage': 43},
    {shopper_id': 'S6', 'coverage': 12},
  ];
  ```
- **How to submit**:
  Complete the source code file named `haversine_coverage.py`.