1) Sketch out a diagram representing the architecture for the following scenario:
   a) receive images from a native mobile app installed in users' mobile device
   b) process such images
   c) store results of the processing stage (point b)
   d) return processed images to users.

2) Describe if you'd make any modification to architecture from Task 1 when you have
   a) 1000 users
   b) 100k users
   c) include timelines it would take to implement functioning architecture for a) & b)

   Assume the scenario from Task 1 happens only once a day. Please make modifications to your sketch as needed. If you're describing use of different tools, describe how you'd implement these tools.

3) Suppose that all the users from 2a and 2b upload their images to the architecture every day twice a day, all at the same time (e.g. 10am and 5pm). Please explain if the architecture proposed at point 2 would be able to correctly process all the requests; if not, please describe what should be added or how the architecture should be modified

4) Coding question
   a) Write the code to implement the section of the architecture responsible for the storage of the information (see point 1a). It is possible to use any kind of programming language and database solution, such as entity-relationship or NoSQL, or a combination of multiple techniques.
   b) List the endpoints of a RESTful API that could be used by a client to interact with the described architecture.
   c) Write the code which would allow the backend to perform the following operations:
      - receive an input image from a client, convert it to grayscale, compute the histogram and store both the original image and the histogram into a database;
      - extract the histograms of the current week for a single user;
      - extract the median histogram of the current day for all users (the output is a single histogram);
   d) Write a piece of code that, given a user id as input, returns n user ids who have the most similar histograms with respect to the input user id.