COMPUTER VISION - LAB 2 HINTS

1. Load into a std::vector< cv::String > the filenames of the calibration images (i.e., images of the checkerboard images) . You may use the OpenCV function:
   cv::utils::fs::glob (const cv::String & directory, const cv::String & pattern,
   std::vector< cv::String > & result, ...);
   with pattern a filter pattern based on '*'/'?' Symbols (e.g., img*.png).
2. Prepare to fill two vectors of vectors:

   1) std::vector<std::vector<cv::Point3f>> points3d;
   2) std::vector<std::vector<cv::Point2f>> points2d;

   that which will contain, **for each calibration image**, the 3D coordinates of the corners (in the chessboard reference system) and the 2D corners extracted in the image, respectively.
3. For each calibration image, find the checkerboard corners using the OpenCV cv::findChessboardCorners() function, and store the found images points into a std::vector<cv::Point2f>. Push back (std::vector::push_back() method) such vector into the points2d vector. Push back into points3d the vector std::vector<cv::Point3f> of 3D points.
4. Compute the camera calibration parameters from points3d and points2d by using the cv:: calibrateCamera(), along with the extrinsic parameters (that is, the position of the camera with respect to each checkerboard view).
5. Print (with names) the parameters with the function std::cout.
6. Compute the mean reprojection error by reprojecting the 3D corners into each calibration image, and comparing the obtained 2D points with the positions of the extracted 2D corners. To reproject the 3D corners, use che cv::: projectPoints () OpenCV functions, with the positions computed by cv::calibrateCamera() (i.e.,. the extrinsic parameters). To compute the mean reprojection, compute the mean Euclidean between reprojected points and extracted corners.
7. Undistort an input test image by using the cv:: initUndistortRectifyMap() and cv::remap() functions.
8. Visualize the input and output images in two different windows by using cv::imshow() function.