Problem 1: Image Publishing & Subscription

In this problem, you need to write some simple image publishing and subscription nodes. *(50 points)*

First create a new package named **imagepub_<roll_number>**. The package would have the following dependencies: `roscpp`, `image_transport`, `cv_bridge`, `dynamic_reconfigure`. Implement the following nodes for this package:

- A node named `cb_publisher`. The node shall advertise an image topic named `cb_img`, containing a generated image of a checkerboard. Your node should have the following ROS parameters:
  - *width* - the width of the advertised image
  - *height* - the height of the advertised image
  - *square size* - the size in pixels of one square of the checkerboard
  - *frequency* - the frequency with which to publish the checkerboard image

  You need to specify default values to these parameters.

  *NOTE: For exercise you are **NOT** allowed to use any opencv functions, including the OpenCV bridge. Instead you should create and manipulate the image message directly.*

- A node named `file_publisher`, that loads an image from disk and advertises it under the topic `image`. You are allowed to use OpenCV for loading the image and the `cv_bridge` for converting an OpenCV image into a ROS image message. Your node should have the following parameters:
  - *file* - the path to the image to load
  - *frequency* - frequency used for publishing the image (use a default parameter here)

  To visualize your results you can use the `image_view` node in the `image_view` package.

You now need to create a new package named **image_enhancer_<roll_number>** for an image processing pipeline in ROS. You will implement nodes that subscribes to an image topic, performs some simple image manipulation, then publishes the resulting image on a separate image topic. Implement the following nodes for this package:

- A node named `image_changer`. The new node should have the following ROS parameters:
  - *brightness* - an integer value greater than or equal to zero
  - *contrast* - a floating point value greater than zero

After subscribing to an image topic, you need to perform brightness and contrast adjustment on incoming image. This is accomplished using a simple linear operation on the raw image data according to the formula:

$$I(x, y)' = c \times I(x, y) + b \tag{1}$$

I(x, y) 0 = c I(x, y) + b where b is the brightness change, c the contrast scaling factor, and $I(x, y)$ the image intensity value at position x, y. For multichannel images, i.e RGB, this operation can be performed on each channel individually.

NOTE: For exercise you are ***NOT*** *allowed to use any opencv functions, including the OpenCV bridge. Instead you should create and manipulate the image message directly.*

To visualize your results you can use the `image_view` node in the `image_view` package.

- A node named `image_dynamic_changer`. This node is similar to the `image_changer` node. However, you are going to make the node respond to parameter changes at runtime using the `dynamic_reconfigure` infrastructure.

  - Create a new directory named **cfg** in the **image_enhancer_<roll_number>** package. Create a new dynamic configuration file in that directory named `ImageChangerConfig.cfg` and make this file executable. The new configuration file should contain the appropriate entries for the `image_dynamic_changer` node.
  - Make the appropriate changes to the `CMakeLists.txt` file for dynamic reconfigure

Test your implementation using the `reconfigure_gui` in the **dynamic reconfigure** package.

Problem 3: Path following using a PD controller

Consider the two-dimensional simple point mass dynamics of a UAV as follows:

$$\dot{x} = v \cos \theta \tag{7}$$

$$\dot{y} = v \sin \theta \tag{8}$$

$$\dot{\psi} = \frac{u}{v} \tag{9}$$

where v is the speed and u is the acceleration produced in along the direction perpendicular to the drone's velocity. You need to develop a MATLAB code to follow a straight-line trajectory using a proportional and derivative (PD) controller. You may choose alternative approaches as well. *(25 points)*