Daily Assignment 20

- *Visualizing rotation about an arbitrary axis:* start from today's practice code, replace the render() function by the one in the next page

- 1. Add **getRotMatFrom(axis, theta)** function
  - *axis* : (unnormalized) rotation axis vector
  - *theta* : rotation angle in degrees
  - Returns the rotation matrix for rotation about *axis* by *theta*
  - **Do not use Rodrigues' rotation formula. You have to use the method described in today's slides from "Let's compute the rotation matrix R"**

- Maybe you'll need
  - The provided normalized() to normalize a vector
  - **a** x **b** (cross product) : np.cross(**a**, **b**)
  - **a** · **b** (inner product) : np.dot(**a**, **b**)
  - To build a matrix using column vectors **a, b, c**: np.column_stack((**a, b, c**))
    - google it for more information
  - Inverse of a matrix **M**: np.linalg.inv(**M**)

Daily Assignment 20

- 2. Add key handling code to change the rotation axis
  - If you **press or repeat** a key, the x, y, z coordinate value of the rotation axis (*gAxis* variable in the code) should be changed as shown in the table:
  - (The rotation axis is already visualized as a white line and initialized to (0,1,0) in the code in the next page)

Key	Transformation
A	Increase x by 0.1
Z	Decrease x by 0.1
S	Increase y by 0.1
X	Decrease y by 0.1
D	Increase z by 0.1
C	Decrease z by 0.1
V	Initialize gAxis with (0,1,0)

```
def getRotMatFrom(axis, theta):
    axis = normalized(axis)
    theta = np.radians(theta)
    p = np.cross(axis, np.array([0.,0.,1.]))
    Raz = np.column_stack((np.array([0.,0.,1.]), p, np.cross(p,
np.array([0.,0.,1.])))) @ np.linalg.inv(np.column_stack((axis, p, np.cross(p,
axis))))
    Rz = np.array([[np.cos(theta), -np.sin(theta), 0],
                   [np.sin(theta), np.cos(theta), 0],
                   [0,0,1]])
    R = np.linalg.inv(Raz) @ Rz @ Raz
    return R

def key_callback(window, key, scancode, action, mods):
    global gCamAng, gCamHeight
    global gAxis
    if action==glfw.PRESS or action==glfw.REPEAT:
        # ...
        elif key==glfw.KEY_A:
            gAxis[0] += .1
        elif key==glfw.KEY_Z:
            gAxis[0] -= .1
        elif key==glfw.KEY_S:
            gAxis[1] += .1
        elif key==glfw.KEY_X:
            gAxis[1] -= .1
        elif key==glfw.KEY_D:
            gAxis[2] += .1
        elif key==glfw.KEY_C:
            gAxis[2] -= .1
        elif key==glfw.KEY_V:
            gAxis = np.array([0.,1.,0.])
```