**به نام خدا**

**سری تمرینات اول**

**استاد مهاجری**

**سبا معدلت**

**تاریخ تحویل: 20/12/1403**

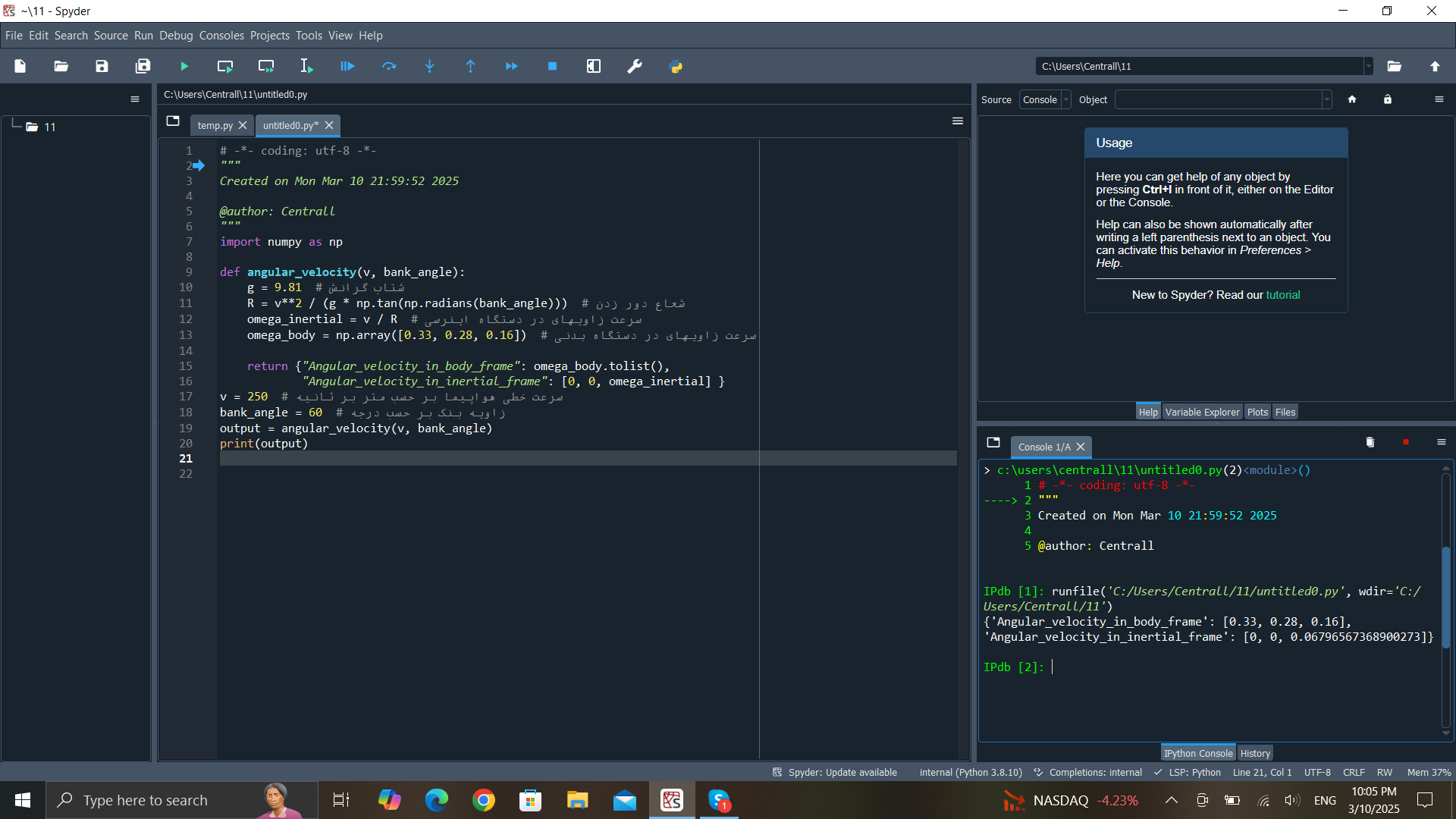
**سوال 1:**

**بردار سرعت زاویه ای هواپیما در دستگاه اینرسی و دستگاه بدنی**

**@author: Centrall"""import numpy as npdef angular\_velocity(v, bank\_angle): g = 9.81 # شتاب گرانش R = v\*\*2 / (g \* np.tan(np.radians(bank\_angle))) # شعاع دور زدن omega\_inertial = v / R # سرعت زاویهای در دستگاه اینرسی omega\_body = np.array([0.33, 0.28, 0.16]) # سرعت زاویهای در دستگاه بدنی return {"Angular\_velocity\_in\_body\_frame": omega\_body.tolist(), "Angular\_velocity\_in\_inertial\_frame": [0, 0, omega\_inertial] }v = 250 # سرعت خطی هواپیما بر حسب متر بر ثانیهbank\_angle = 60 # زاویه بنک بر حسب درجهoutput = angular\_velocity(v, bank\_angle)print(output)**

**خروجی:**

**{'Angular\_velocity\_in\_body\_frame': [0.33, 0.28, 0.16], 'Angular\_velocity\_in\_inertial\_frame': [0, 0, 0.06796567368900273]}**

****

**سوال3: ماتریس انتقال و محاسبه کواترنیون و زوایای اویلر و بردار دوران**

**Created on Mon Mar 10 22:06:40 2025@author: Centrall"""import numpy as npfrom scipy.spatial.transform import Rotation as Rdef check\_rotation\_matrix(matrix): # الف) if not np.allclose(np.dot(matrix, matrix.T), np.eye(3)): return "ماتریس ورودی شرایط یک ماتریس انتقال را ندارد" # ب) کواترنیون، زوایای اویلر و بردار دوران rotation = R.from\_matrix(matrix) quaternion = rotation.as\_quat() euler\_angles = rotation.as\_euler('xyz', degrees=True) rotation\_vector = rotation.as\_rotvec() return {"Euler\_angles": euler\_angles.tolist(), "Quaternion\_vector": quaternion.tolist(), "Rotation\_vector": rotation\_vector.tolist()}C\_b\_i = np.array([[0.2802, 0.1387, 0.9499], [0.1962, 0.9603, -0.1981], [-0.9397, 0.2418, 0.2418]])output = check\_rotation\_matrix(C\_b\_i)print(output)**

**خروجی:**

**runfile('C:/Users/Centrall/11/untitled2.py', wdir='C:/Users/Centrall/11')ماتریس ورودی شرایط یک ماتریس انتقال را ندارد**

**بوناس:**

@author: Centrall"""import numpy as npdef flight\_matrix(omega, flight\_phase): p, q, r = omega if flight\_phase == "Cruise": # پرواز Cruise f = np.array([ [0, -r, q], [r, 0, -p], [-q, p, 0] ]) elif flight\_phase == "Pull-up": # پرواز Pull-up f = np.array([ [0, -r, q], [r, 0, -p], [-q, p, 0] ]) elif flight\_phase == "Coordinated-turn": # پرواز Coordinated-turn f = np.array([ [0, -r, q], [r, 0, -p], [-q, p, 0] ]) else: raise ValueError("فاز پروازی نامعتبر است") return fomega = [0.33, 0.28, 0.16]flight\_phase = "Cruise"output= flight\_matrix(omega, flight\_phase)print(output)

خروجی:

runfile('C:/Users/Centrall/11/untitled3.py', wdir='C:/Users/Centrall/11')[[ 0. -0.16 0.28] [ 0.16 0. -0.33] [-0.28 0.33 0. ]]

