Siddharth Cherukupalli

Dr. Nitin Sanket

FIRE198 - Autonomous Unmanned Systems

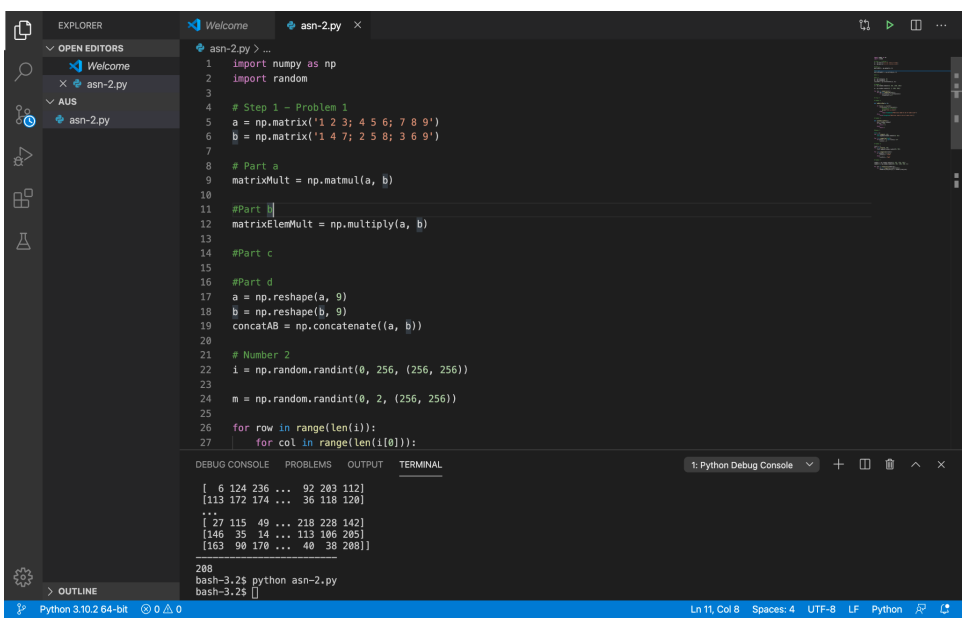
14 February 2022

ASN2: Let's Get Pythonic!

1. We need __name__ == "__main__" because without it, it will open a new module with different functions. Having the __main__ makes sure that this module is being run by the user and therefore we can do appropriate actions in the module that we're in.
4. To convert a list to a tuple, use tuple(list_name). To convert a list to a numpy array, use numpy.array(list_name). To convert a list to a dictionary, use dict(zip(list_1, list_2)).

What IDE/Editor did you choose in step 0 and why?

I used Visual Studio Code for coding and I did so because it is an editor I am already familiar with and comfortable with. I have used it for other Python projects that I worked on so I decided to go ahead and use this.



Code - Sample Inputs and Outputs for Step 1:

```python
# Step 1 - Problem 1
a = np.matrix('1 2 3; 4 5 6; 7 8 9')
b = np.matrix('1 4 7; 2 5 8; 3 6 9')

# Part a
matrixMult = np.matmul(a, b)
print(matrixMult)

#Part b
matrixElemMult = np.multiply(a, b)
print(matrixElemMult)

#Part c
transposeA = np.matrix.transpose(a)
inverseA = inv(a)
prod = np.dot(transposeA, b)
finalProd = np.dot(prod, inverseA)
print(finalProd)

#Part d
a = np.reshape(a, 9)
b = np.reshape(b, 9)
cv = np.concatenate((a, b))

l2norm = np.linalg.norm(cv, axis=0)

print(cv)
print(l2norm)
```

```
bash 5.2$ python ash 2.py
[[ 14  32  50]
 [ 32  77 122]
 [ 50 122 194]]
[[ 1  8 21]
 [ 8 25 48]
 [21 48 81]]
[[ 56. -62.  31.]
 [ 88.  74. -37.]
 [120.  82.  23.]]
[[1 2 3 4 5 6 7 8 9]
 [1 4 7 2 5 8 3 6 9]]
[ 1.41421356  4.47213595  7.61577311  4.47213595  7.07106781 10.
  7.61577311 10.         12.72792206]
```

Code - Sample Inputs and Outputs for Step 2:

```python
ash-2.py > ...
# Number 2
i = np.random.randint(0, 256, (256, 256))

m = np.random.randint(0, 2, (256, 256))

for row in range(len(i)):
    for col in range(len(i[0])):
        if (i[row][col] == m[row][col]):
            i[row][col] = 0

# Step 2

# Number 2
def sqMatrixOps(a, b):

    if (len(a) == len(b)):
        if(len(a[0]) == len(b[0])):
            transposeA = np.matrix.transpose(a)
            inverseA = inv(a)
            prod = np.dot(transposeA, b)
            finalProd = np.dot(prod, inverseA)
            return finalProd
        else:
            raise Exception("Matrices need to be of same size!")
    else:
        raise Exception("Matrices need to be of same size!")

print(sqMatrixOps(a, b))

# Number 3
def probabilityFunc():
    num = random.random()
    if num < 0.5:
        return -1
    else:
        return 1
```

```python
print(probabilityFunc())

#Number 5
list = []
for i in range(0, 10):
    list.append(random.randint(0, 10))

tempList = []
for i in range(11):
    tempList.append(i)

list = [0 if (x>=5 and x <=7) else 1 for x in tempList]

print(list)

# Number 6
list2 = []
for i in range(0, 10):
    list2.append(random.randint(0, 10))

list2 = ["Even" if x == x % 2 else "Odd" for x in tempList]


print(list2)

# Number 7
randArr = np.random.randint(0, 255, (256, 256))
randArrS = np.random.randint(0, 255, (256, 256, 3))

for row in range(len(randArrS)):
    for col in range(len(randArrS[0])):
        randArrS[row][col][2] = randArr[row][col]

print(randArr)
print(randArrS)
```

```
1
[1, 1, 1, 1, 1, 0, 0, 0, 1, 1, 1]
['Even', 'Even', 'Odd', 'Odd', 'Odd', 'Odd', 'Odd', 'Odd', 'Odd', 'Odd', 'Odd']
[[ 65  74  45 ... 228  69   8]
 [ 54 182 196 ...  44 237 152]
 [120  85 145 ...  34 144 187]
 ...
 [210  12   7 ... 146 106 100]
 [215  49 127 ... 242 115 161]
 [115 101 172 ... 252  40  77]]
[[[202 124  65]
  [139  61  74]
  [211 194  45]
  ...
  [178 205 228]
  [215  87  69]
  [ 50 111   8]]

 [[250 204  54]
  [230 172 182]
  [154 136 196]
  ...
  [135 218  44]
  [  9  42 237]
  [114 130 152]]

 [[207  10 120]
  [188 141  85]
  [101  72 145]
  ...
  [240 137  34]
  [107 141 144]
  [  2 141 187]]

 ...

 [[ 25 204 210]
  [235 167  12]
  [ 80 188   7]
  ...
  [ 27  64 146]
  [131  44 106]
  [  9 246 100]]
```

Challenges you faced in solving this assignment along with lessons learned.

Some challenges I faced included finding out how to transpose a matrix as well as other matrix operations as I didn't know it prior to this assignment. Using Google and resources like Stack Overflow, I was able to figure out the solutions to those problems. I learned how to better

visualize a 3d space, as we were assigned to make a 256 x 256 x 3 array, which forced me to think in a different way since we're more accustomed to 2d spaces.