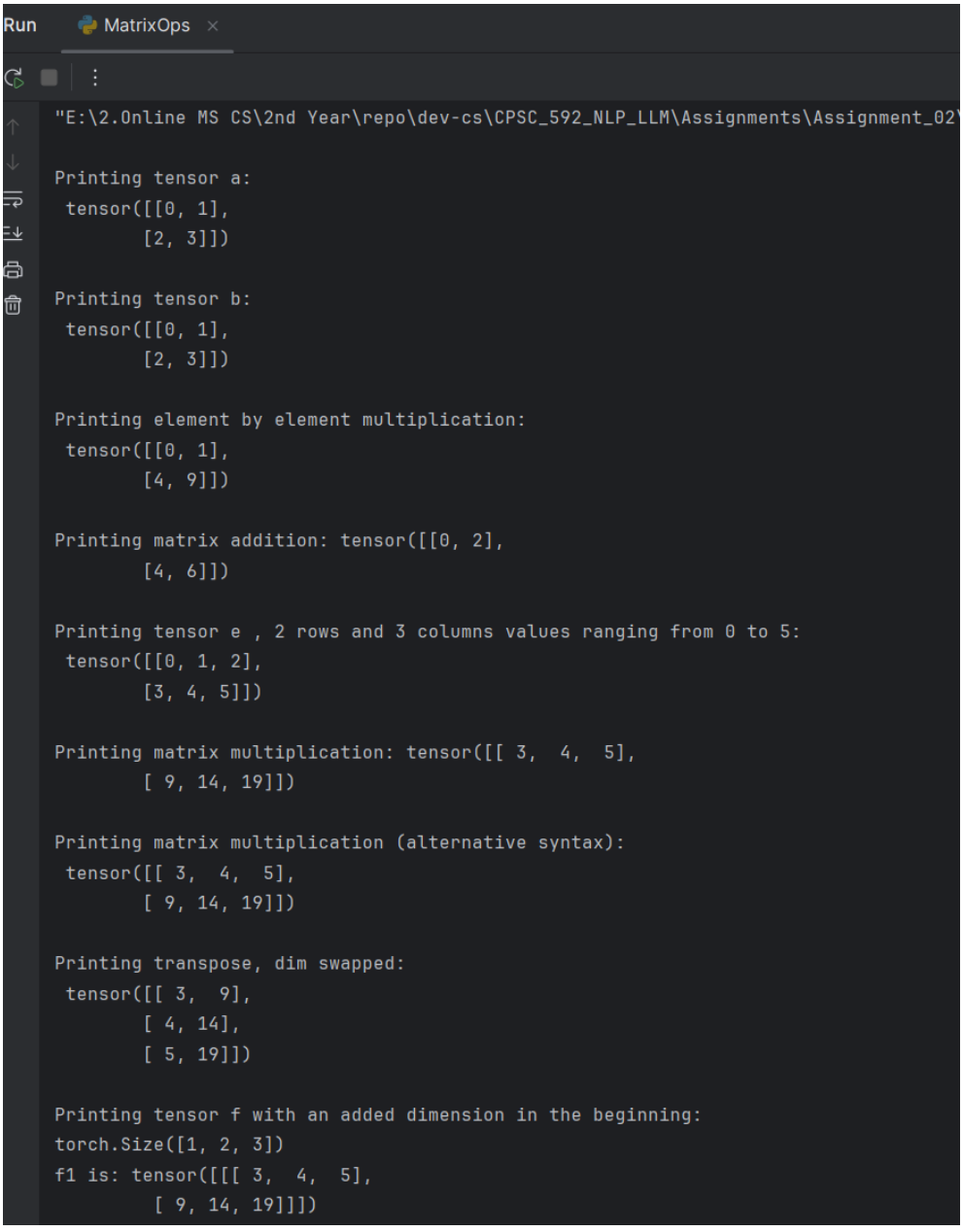
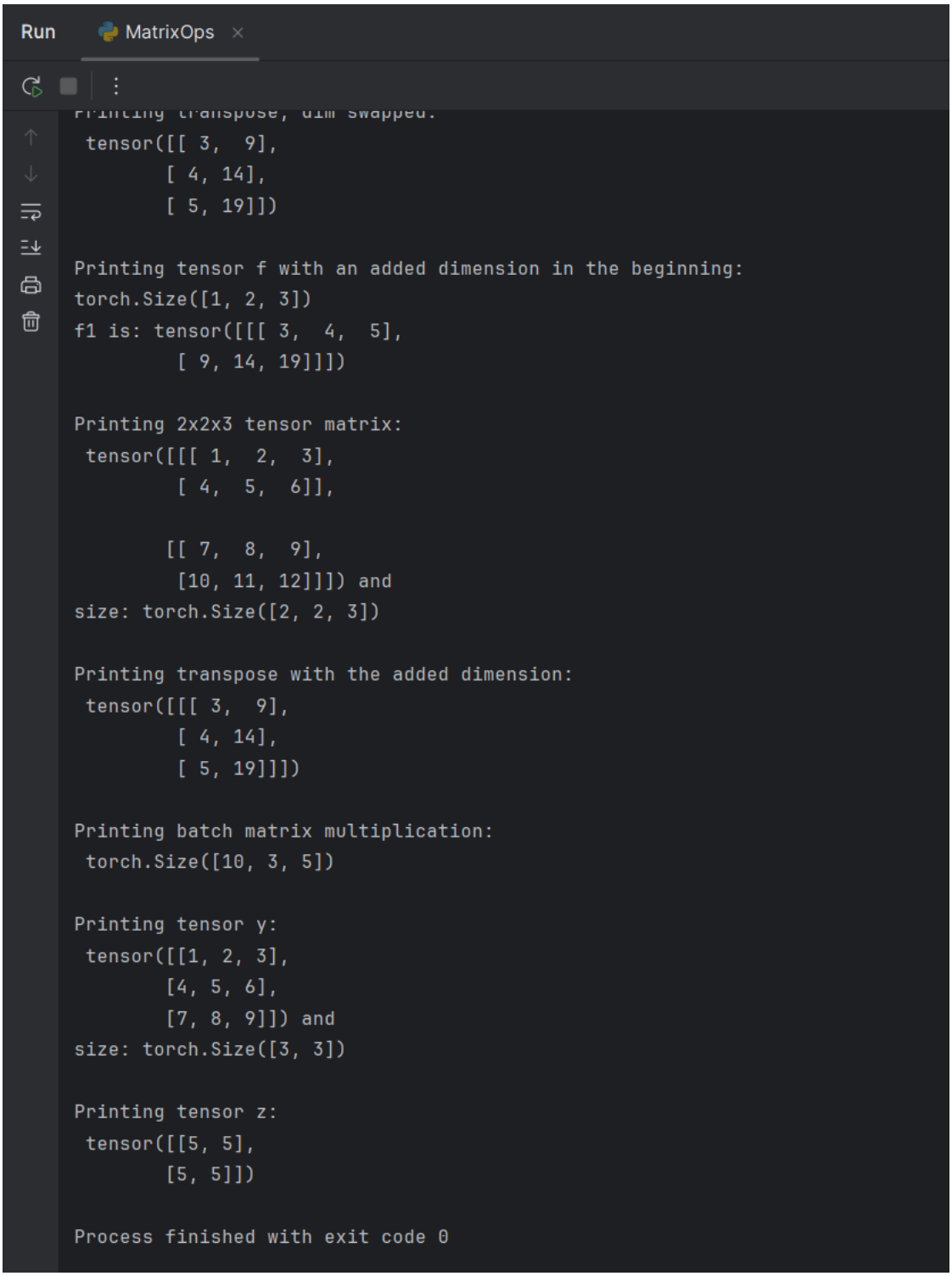
Assignment 02 (St ID: 1174066)

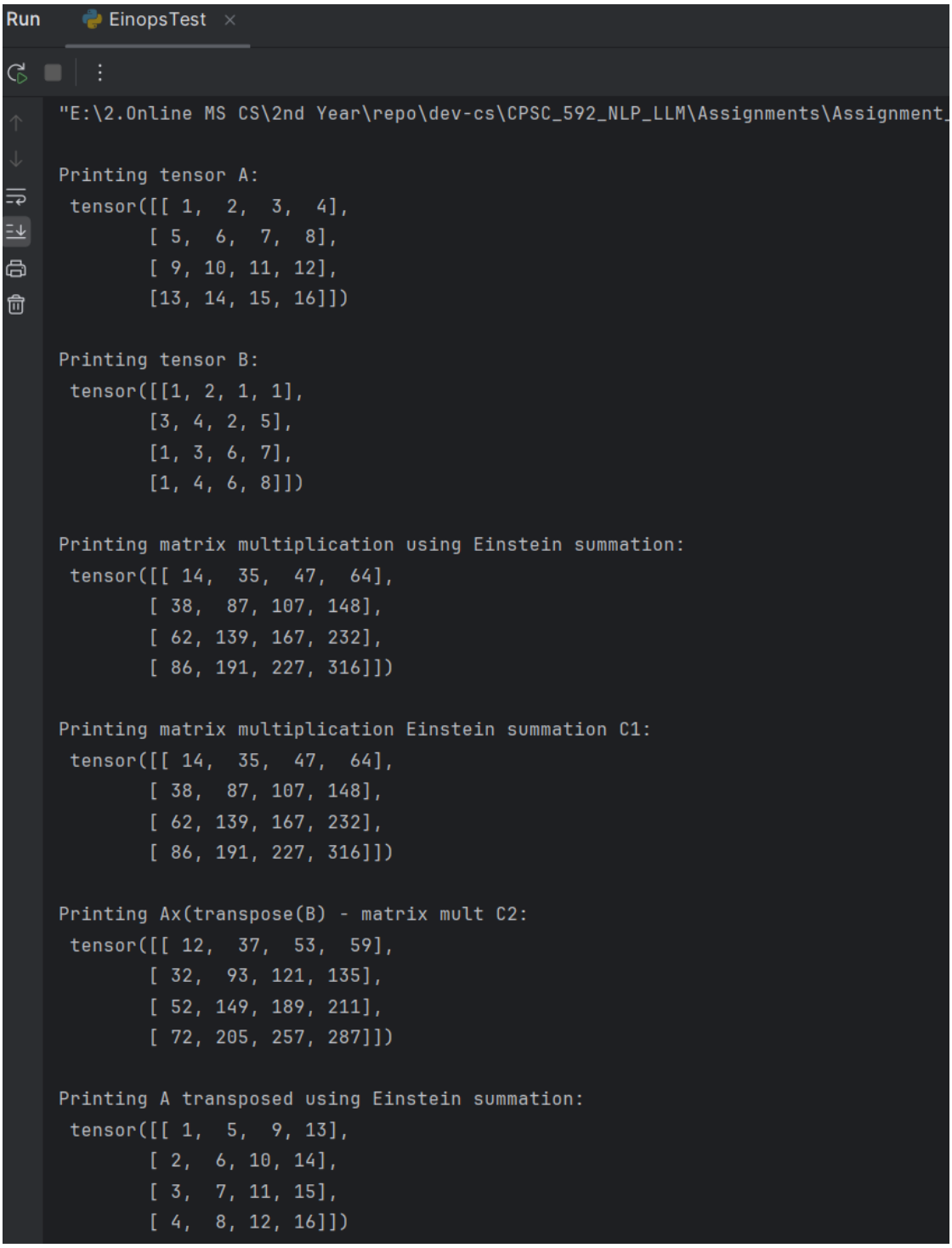
OUTPUTS

MatrixOps.py



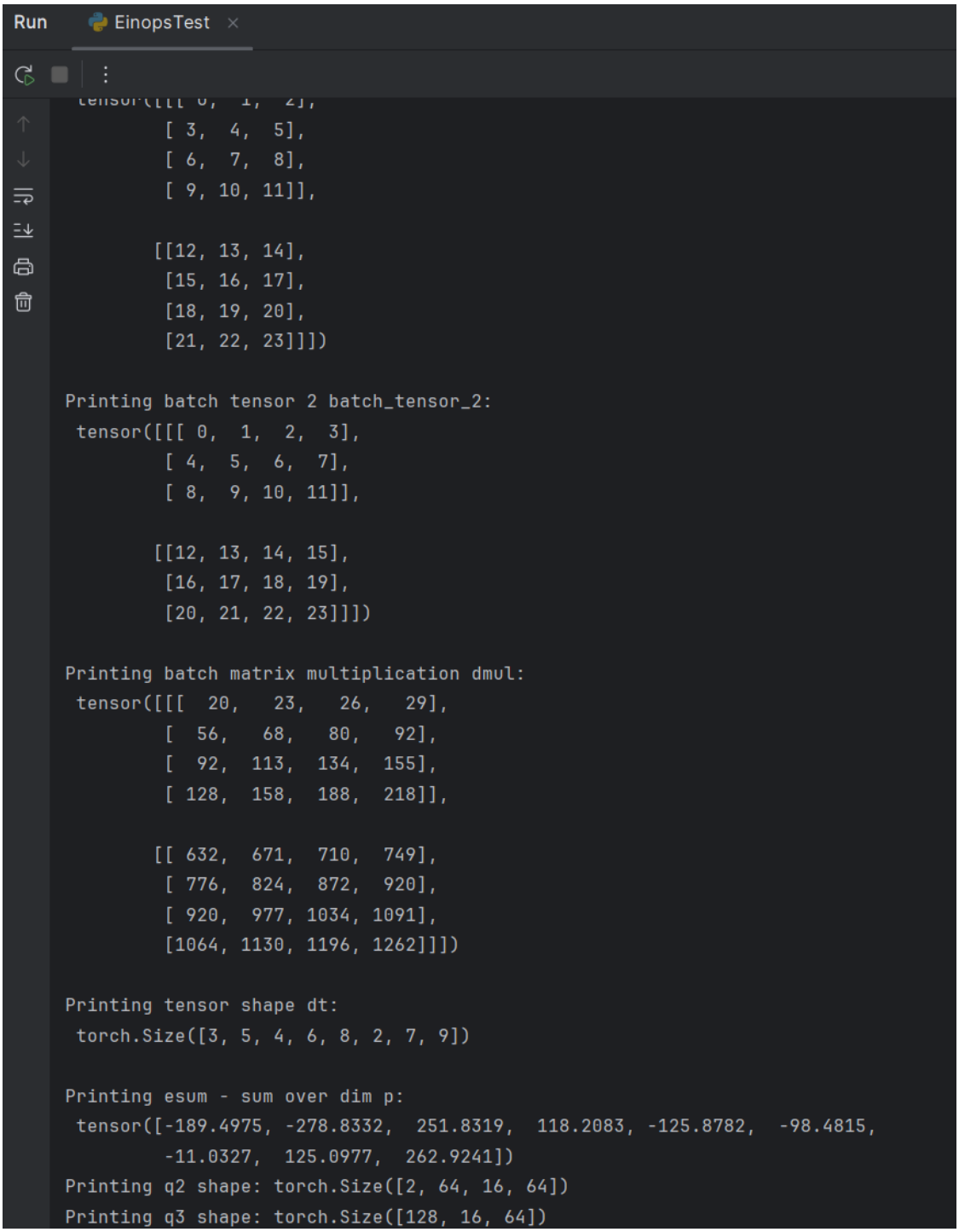

Run
MatrixOps
Printing transpose, dim swapped:
tensor([[3, 9],
[4, 14],
[5, 19]])
Printing tensor f with an added dimension in the beginning:
torch.Size([1, 2, 3])
f1 is: tensor([[[3, 4, 5],
[9, 14, 19]]])
Printing 2x2x3 tensor matrix:
tensor([[[1, 2, 3],
[4, 5, 6]],
[[7, 8, 9],
[10, 11, 12]]]) and
size: torch.Size([2, 2, 3])
Printing transpose with the added dimension:
tensor([[[3, 9],
[4, 14],
[5, 19]]])
Printing batch matrix multiplication:
torch.Size([10, 3, 5])
Printing tensor y:
tensor([[1, 2, 3],
[4, 5, 6],
[7, 8, 9]]) and
size: torch.Size([3, 3])
Printing tensor z:
tensor([[5, 5],
[5, 5]])
Process finished with exit code 0

EinopsTest.py



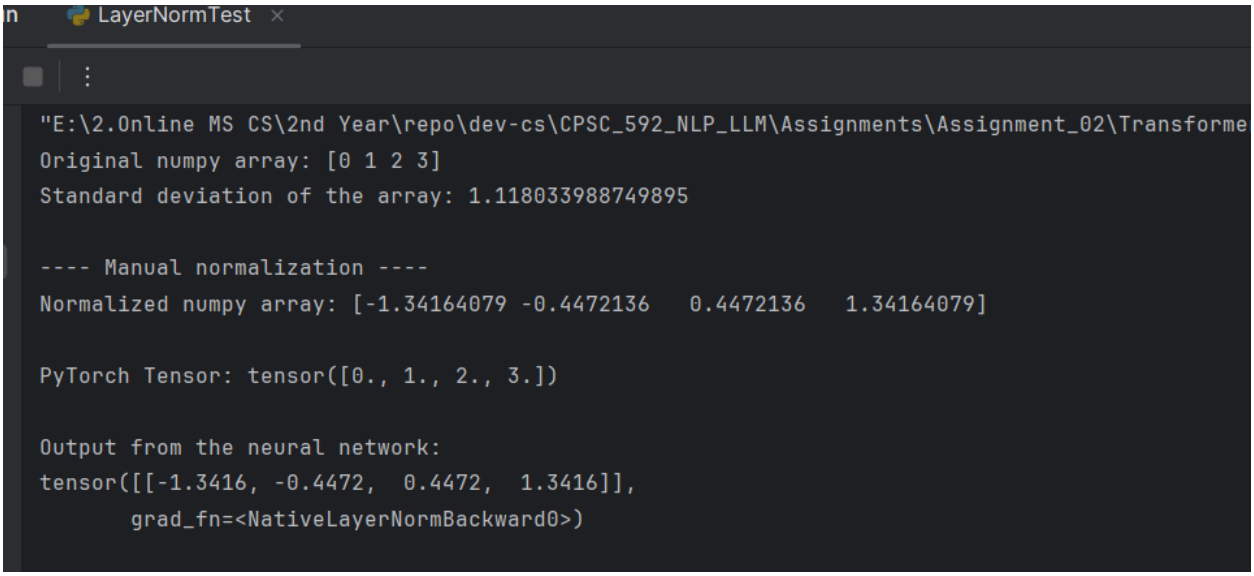
Run EinopsTest

```
 tensor([ 1,  6, 11, 16])

Printing sum diagonal elements - trace C4:
 tensor(34)

Printing sum column elements (row wise sum) C5:
 tensor([28, 32, 36, 40])

Printing element wise product C6:
 tensor([[  1,   4,   3,   4],
        [ 15,  24,  14,  40],
        [  9,  30,  66,  84],
        [ 13,  56,  90, 128]])

Printing cube elements C7:
 tensor([[   1,    8,   27,   64],
        [ 125,  216,  343,  512],
        [ 729, 1000, 1331, 1728],
        [2197, 2744, 3375, 4096]])

Printing transpose C8:
 tensor([[ 1,  5,  9, 13],
        [ 2,  6, 10, 14],
        [ 3,  7, 11, 15],
        [ 4,  8, 12, 16]])

Printing multiply row wise and add each row C9:
 tensor([ 12,  93, 189, 287])

Printing outer product douter:
 tensor([[ 3,  6,  9, 12],
        [ 5, 10, 15, 20],
        [ 7, 14, 21, 28],
        [ 9, 18, 27, 36]])

Printing inner product dinner:
 tensor(70)
```

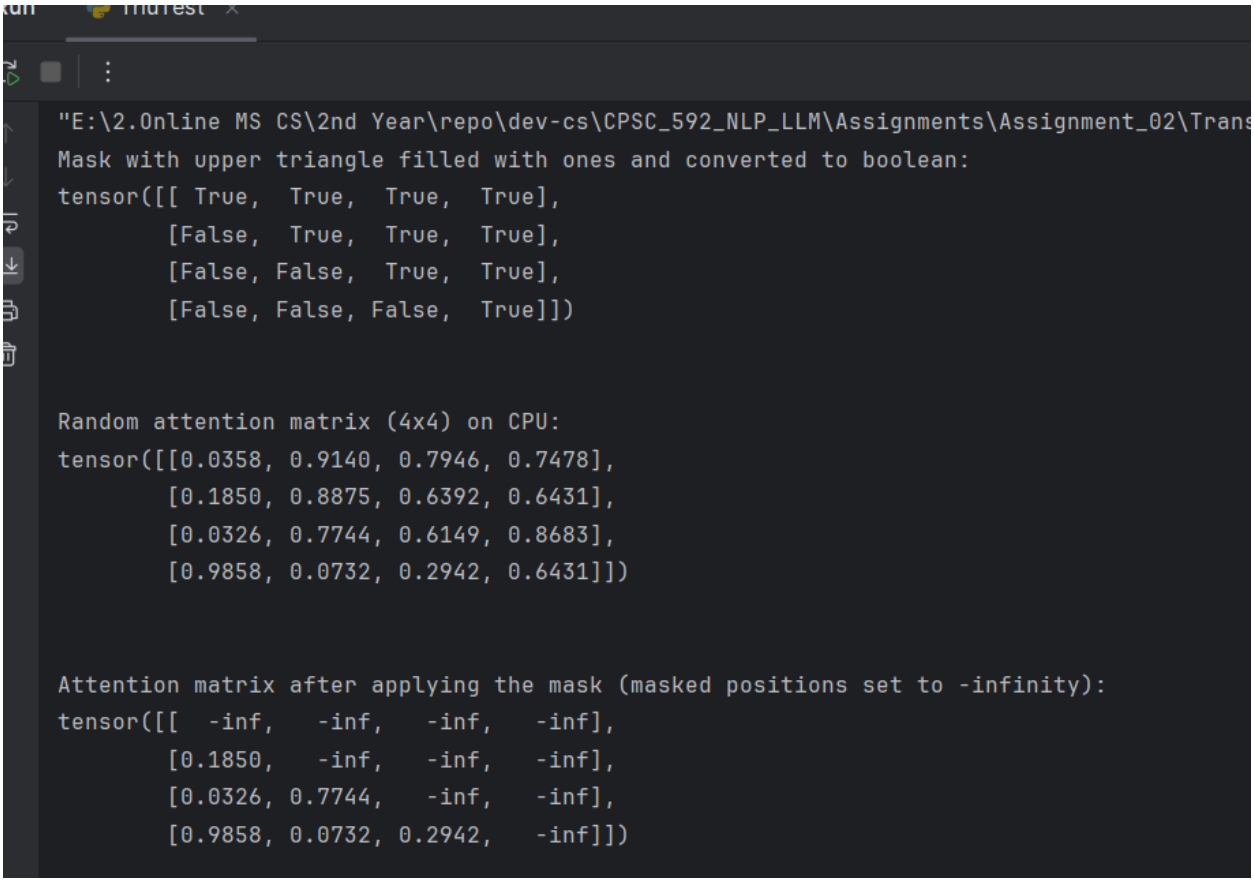

Run
EinopsTest
tensor([[[0, 1, 2],
[3, 4, 5],
[6, 7, 8],
[9, 10, 11]],
[[12, 13, 14],
[15, 16, 17],
[18, 19, 20],
[21, 22, 23]]])
Printing batch tensor 2 batch_tensor_2:
tensor([[[0, 1, 2, 3],
[4, 5, 6, 7],
[8, 9, 10, 11]],
[[12, 13, 14, 15],
[16, 17, 18, 19],
[20, 21, 22, 23]]])
Printing batch matrix multiplication dmul:
tensor([[[20,   23,   26,   29],
[56,   68,   80,   92],
[92, 113, 134, 155],
[128, 158, 188, 218]],
[[632, 671, 710, 749],
[776, 824, 872, 920],
[920, 977, 1034, 1091],
[1064, 1130, 1196, 1262]]])
Printing tensor shape dt:
torch.Size([3, 5, 4, 6, 8, 2, 7, 9])
Printing esum - sum over dim p:
tensor([-189.4975, -278.8332, 251.8319, 118.2083, -125.8782, -98.4815,
-11.0327, 125.0977, 262.9241])
Printing q2 shape: torch.Size([2, 64, 16, 64])
Printing q3 shape: torch.Size([128, 16, 64])

TransformerLayer.py

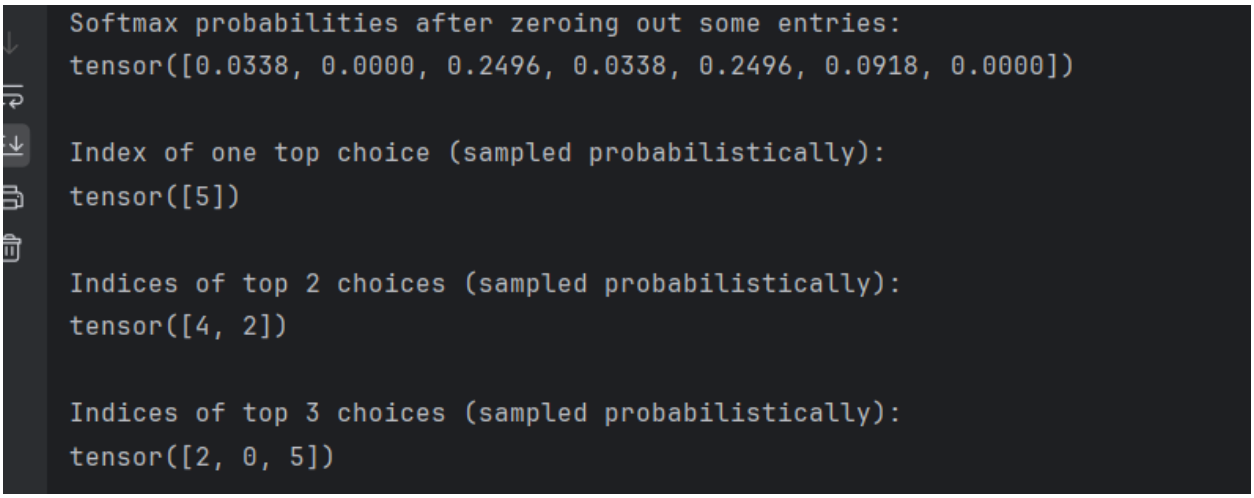
LayerNormTest.py



TriuTest.py



MultinomialTest.py



CrossEntropyTest.py

```
tensor([[0, 1, 2, 3, 4, 5, 6, 7, 8, 9]])
tensor([[7, 8, 9]])
-----logits--------
tensor([[1., 3., 5.],
        [2., 4., 1.]])

cross entropy loss by pytorch= tensor(1.1564)

-----softmax------
tensor([[0.0159, 0.1173, 0.8668],
        [0.1142, 0.8438, 0.0420]])

nll loss by pytorch = tensor(1.1564)

cross entropy by our calculation= tensor(1.1564)
```

KLDivergenceTest.py