Assignment1

October 2, 2018

Problem number 1

```
In [1]: import numpy as np
        import math as ma
        array1= np.arange(1,11,1)
        multTable= np.outer(array1,array1)
        print(multTable)
```

```
[[  1   2   3   4   5   6   7   8   9  10]
 [  2   4   6   8  10  12  14  16  18  20]
 [  3   6   9  12  15  18  21  24  27  30]
 [  4   8  12  16  20  24  28  32  36  40]
 [  5  10  15  20  25  30  35  40  45  50]
 [  6  12  18  24  30  36  42  48  54  60]
 [  7  14  21  28  35  42  49  56  63  70]
 [  8  16  24  32  40  48  56  64  72  80]
 [  9  18  27  36  45  54  63  72  81  90]
 [ 10  20  30  40  50  60  70  80  90 100]]
```

problem number 2

```
In [2]: array1=multTable[3:7,3:7]
        print(array1)
```

```
[[16 20 24 28]
 [20 25 30 35]
 [24 30 36 42]
 [28 35 42 49]]
```

problem number 3

```
In [3]: array2= np.arange(1,17).reshape(4,4)
        print(array2)
```

```
[[ 1  2  3  4]
 [ 5  6  7  8]
 [ 9 10 11 12]
 [13 14 15 16]]
```

problem number 4

```
In [4]: output= array1%2==0
        print(output)
```

```
[[ True  True  True  True]
 [ True False  True False]
 [ True  True  True  True]
 [ True False  True False]]
```

problem number 5

```
In [5]: numEvens= np.sum(output==True, axis= 1)
        print(np.sum(numEvens))
```

```
12
```

problem number 6

```
In [6]: print(np.sqrt(array1))
```

```
[[4.         4.47213595 4.89897949 5.29150262]
 [4.47213595 5.         5.47722558 5.91607978]
 [4.89897949 5.47722558 6.         6.4807407 ]
 [5.29150262 5.91607978 6.4807407  7.        ]]
```

problem number 7

```
In [7]: numsToAdd= np.diag([1,1,1,1,1,1,1,1,1,1])
        print(multTable+numsToAdd)
```

```
[[  2   2   3   4   5   6   7   8   9  10]
 [  2   5   6   8  10  12  14  16  18  20]
 [  3   6  10  12  15  18  21  24  27  30]
 [  4   8  12  17  20  24  28  32  36  40]
 [  5  10  15  20  26  30  35  40  45  50]
 [  6  12  18  24  30  37  42  48  54  60]
 [  7  14  21  28  35  42  50  56  63  70]
 [  8  16  24  32  40  48  56  65  72  80]
 [  9  18  27  36  45  54  63  72  82  90]
 [ 10  20  30  40  50  60  70  80  90 101]]
```

problem number 8

```
In [8]: out= np.flip(multTable, axis=-1)
        out= np.flip(out, axis=0)
        print(out)
```

```
[[100  90  80  70  60  50  40  30  20  10]
 [ 90  81  72  63  54  45  36  27  18   9]
 [ 80  72  64  56  48  40  32  24  16   8]
 [ 70  63  56  49  42  35  28  21  14   7]
 [ 60  54  48  42  36  30  24  18  12   6]
 [ 50  45  40  35  30  25  20  15  10   5]
 [ 40  36  32  28  24  20  16  12   8   4]
 [ 30  27  24  21  18  15  12   9   6   3]
 [ 20  18  16  14  12  10   8   6   4   2]
 [ 10   9   8   7   6   5   4   3   2   1]]
```

question number 9

```
In [9]: randomsArray= np.random.uniform(0.0,10.0,(100))
        randomsArray= np.sort(randomsArray, axis=-1)
        randomsArray= randomsArray.reshape((10,10))
        print(randomsArray)
```

```
[[0.04831699 0.06977811 0.17948223 0.18146823 0.19263135 0.21120756
  0.28753959 0.3136819  0.35531047 0.64938249]
 [0.64941136 0.6526662  0.65641655 0.73728763 0.83321434 1.14642381
  1.28430999 1.32285251 1.34556325 1.41352289]
 [1.51958926 1.62477506 1.77101419 1.84150758 1.94496302 1.95448319
  2.11524271 2.15042862 2.19167563 2.22031356]
 [2.23327877 2.29909724 2.39229898 2.44186154 2.56448532 2.68888087
  2.77739038 2.87635949 3.1494188  3.18449139]
 [3.20396768 3.27786194 3.70946014 4.02148889 4.02671643 4.19372436
  4.41209345 4.42792126 4.56187084 4.61738988]
 [4.66936187 4.67603619 4.83263023 4.9144704  4.99528819 5.09874294
  5.13285379 5.1836813  5.26397591 5.47283817]
 [5.55288763 5.5531319  5.56137349 5.63355933 5.69856914 5.92173343
  5.97147497 6.07319601 6.52359269 6.55669774]
 [6.73988182 6.84955868 7.01207265 7.24808608 7.25716649 7.47277347
  7.53370166 7.54407497 7.89446961 7.98858857]
 [8.09112368 8.15899417 8.31243391 8.49799207 8.69069551 8.75500915
  8.76367422 8.81749508 8.86761091 8.9116446 ]
 [8.94296287 8.96974189 9.08581633 9.21663652 9.34731945 9.4002021
  9.41712314 9.43632528 9.55384899 9.9755721 ]]
```

problem number 10

```
In [10]: def getAvg(inputArray):
             return np.sum(inputArray, axis=1)/10.0
         #output= np.vectorize(getAvg)(multTable)
         print(getAvg(multTable))
```

```
[ 5.5 11.  16.5 22.  27.5 33.  38.5 44.  49.5 55. ]
```

question number 11

```
In [11]: randomsArray= np.random.randint(0,50,10)
         print('randoms array: ',randomsArray)
         def mergeSort(arrayA):
             if(len(arrayA)>1):
                 b,c=np.split(arrayA,[ma.floor(len(arrayA)/2.0)])
                 b=mergeSort(b)
                 c=mergeSort(c)
                 arrayToBeSorted=np.hstack([b,c])
                 return merge(arrayToBeSorted,0,len(arrayToBeSorted)-1)
             else:
                 return arrayA
         def merge(arrayA,l,r):
             if(l<r):
                 s= ma.floor(len(arrayA)/2)
                 output= np.partition(arrayA,s)
                 b,c=np.split(output,[s])
                 d=np.array(output[s])
                 b=merge(b,l,l+s-1)
                 c=merge(c,l+s,r)
                 output= np.hstack([b,c])
                 return output
             return arrayA
         print(mergeSort(randomsArray))
```

```
randoms array:  [15 23 12  6 20  4  8 30 45  7]
[ 4  6  7  8 12 15 20 23 30 45]
```