

Bhagirath Tallapragada Student ID: 002575358 email: [btallapragada1@student.gsu.edu](mailto:btallapragada1@student.gsu.edu)

1. Write a function that finds the factorial of a number. Test the function finding the factorial of the number 10

```
1 def fact(x):
2     if(x==0 or x==1):
3         return 1
4     else:
5         return x*fact(x-1)
```

```
1 print(fact(10))
```

3628800

2. Write a function that determines if a number is a prime number. Test the function with the following numbers: 3 and 1251.

```
1 def isprime(x):
2     if(x<=1):
3         return False
4     i=2
5     while(i<x):
6         if(x%i==0):
7             return False
8         i+=1
9     return True
```

```
1 print(isprime(3))
```

True

```
1 print(isprime(1251))
```

False

3. Write a function that transposes a matrix. Test with a 3 x 3 identity matrix

```
1 def transp(x):
2     y=[[x[j][i] for j in range(len(x))] for i in range(len(x[0]))]
3     return y
```

```

1 a=[[1,0,0],[0,1,0],[0,0,1]]
2 b=transp(a)
3 for i in b:
4     print(i)

[1, 0, 0]
[0, 1, 0]
[0, 0, 1]

```

4. Using Pandas and Numpy, write a piece of code to combine two series into one. Test your code with: `np.random.seed(0)` `series1 = pd.Series(np.arange(10))` `series2 = pd.Series(np.arange(26))` Do not forget to set the seed, or your answer won't be reproducible.

```

1 import numpy as np
2 import pandas as pd
3
4 np.random.seed(0)
5 series1 = pd.Series(np.arange(10))
6 series2 = pd.Series(np.arange(26))
7
8 print(series1)
9 print(series2)
10
11 mergedseries=series1.append(series2)
12 print('printing combined series')
13 print(mergedseries)

```

```

0    0
1    1
2    2
3    3
4    4
5    5
6    6
7    7
8    8
9    9
dtype: int64
0    0
1    1
2    2
3    3
4    4
5    5
6    6
7    7
8    8
9    9
10   10
11   11

```

```

12    12
13    13
14    14
15    15
16    16
17    17
18    18
19    19
20    20
21    21
22    22
23    23
24    24
25    25
dtype: int64
printing combined series
0      0
1      1
2      2
3      3
4      4
5      5
6      6
7      7
8      8
9      9
0      0
1      1
2      2
3      3
4      4
5      5
6      6
7      7
8      8
9      9

```

5. Using pandas, write a piece of code to convert the first character of each element in a series to uppercase. Test your code with: `series = pd.Series(['nlp', 'will', 'be', 'easy?'])`

```

1 series = pd.Series(['nlp', 'will', 'be', 'easy?'])
2
3 def supper(series):
4     updated=[]
5     for i,v in series.iteritems():
6         #print(v)
7         ele=str(v[0]).upper()
8         ele=ele+v[1:len(x)+1]
9         #print(x[1:len(x)])
10    #print(ele)
11    updated+= [ele]
12    return pd.Series(updated)
13

```

```

1 x=supper(series)
2 print(x)

0      Nlp
1      Will
2       Be
3    Easy?
dtype: object

```

Bonus: Using pandas, write a piece of code to get the frequency of unique values in the entire dataframe (not using built in functions). Test your code with: `dataFrame = pd.DataFrame(np.random.randint(5, 10, 20).reshape(-1, 4), columns = list('abcd'))`

```

1 dataFrame = pd.DataFrame(np.random.randint(5, 10, 20).reshape(-1, 4), columns = list('abcd'))
2 print(dataFrame)

```

```

      a  b  c  d
0  9  8  7  5
1  6  6  8  5
2  5  6  7  9
3  7  5  8  7
4  7  5  6  5

```

```

1 def df_unique(df):
2     keyval={}
3     dlist=list(df.values.flatten())
4     for i in dlist:
5         if i in keyval.keys():
6             keyval[i]+=1
7         else:
8             keyval[i]=1
9
10    return len(list(keyval.keys()))
11 # the function also stores the number of occurrences of each key in the corresponding value

```

```

1 unique=df_unique(dataFrame)
2 print(unique)

```

5

