PS 1 Solutions

Computer for Sciences SCI 378A

Your turn 3A

(a)

```
# for loop
import numpy as np
for x in np.arange(0, 10, 0.1):
    y = np.exp(-(x**2))
    print(y)
```

```
#vectorize
import numpy as np
x = np.arange(0, 10, 0.1)
y = np.exp(-(x**2))
y
```

PS 1 Solutions:

(6) # without importing factorials

```
import numpy as np
N = 10
mu = 2.0

for n in np.arange(0, N+1, 1):
    if (n==0):
            y = np.exp(-mu) * (mu **n)
            print("n = {}, y={:.4f}".format(n, y))

    else:
            p = np.arange(1, n+1, 1)
            y = (np.exp(-mu)*(mu **n))/p.prod()
            print("n = {}, p={}, y= {:.6f}".format(n,p,y))
```

PS1 C

```
# using imported factorial from scipy.special

import numpy as np
from scipy.special import factorial

N =10
mu = 2.0

n = np.arange(0, N+1, 1)
y = np.exp(-mu)* mu **n / factorial(n)

# optional print statement

print(n, y)
```