Q.1 (a) If we defined the following function:

*def doit(inum):*

*if (inum == 0) : return 1*

*return doit(inum – 1) \* inum*

the call to *doit(4)* would return: \_24\_\_\_\_\_\_\_\_

Q.1 (b) That question was (pick one):

Pretty easy

Q.2. NP stands for (pick only one):

Nondeterministic Polynomial

Q.3. The result of flipping a fair coin follows a

I couldn’t remember so I looked it up

Bernoulli distribution

But I don’t Think I ever learned that

Q.4. The derivative (x3)’ =

I don’t know.

Q.5. The transpose of:

\left[ \begin{array}{cc}
1 & 2 \\
3 & 4 \\
5 & 6
\end{array} \right]

is

[

[1,3,5]

[2,4,6]

]

Q.6. (a) The function:

sum(i\*(i+1) for i in range(1,4))

= \_\_\_\_\_\_\_

20

Q.6 (b) How will you change the function doit() you have seen before to compute this new function? Fill in the two blanks.

def doit(inum):

if (inum == 0) : return \_\_\_?\_\_\_\_

return doit(inum – 1) \_\_\_\_\_?\_\_\_\_\_\_\_\_\_

Q.7. Let P(X) = 0.1, P(Y) = 0.2, P(X|Y) = 0.4. What is P(Y|X)?

pY|X = (pX|Y \*pX)/pY

pX|Y =0.2

#### Q.8. Write a program to find the five character-level trigrams (strings with three characters, such as "abc" or "rey") that appear the highest number of times in the following poem (“When You Are Old” by W. B. Yeats). Please lowercase all letters. The trigrams should not contain spaces, but may include punctuations. The result should list the top five trigrams and how many times they occur, in the decreasing order of the occurrence frequency.

*poem = poem.lower().replace("\n"," ")*

*print poem[-10:]*

*count = Counter("".join(i) for i in zip(poem,poem[1:],poem[2:]) if " " not in i)*

*print count.most\_common(5)*

***[('and', 12), ('you', 8), ('the', 8), ('ove', 7), ('lov', 6)]***