Below are sample questions to help you prepare for Exam #3. Make sure you can solve all of these problems by hand. For most of the questions, you can check your answers by typing in the programs and seeing what happens on the computer. When writing your programs by hand, please write clearly and make sure that the indenting of your program is clear.

1. Consider the following Python program.

```python
fruits = ['cherry', 'pecan', 'apple']
print [fruit + ' pie' for fruit in fruits]  # Line A
vec = [2, 4, 6]
print [3*x for x in vec]  # Line B
print [3*x for x in vec if x > 3]  # Line C
print [[x, x**2] for x in vec]  # Line D
vec1 = [2, 4, 6]
vec2 = [4, 3, -9]
print [x*y for x in vec1 for y in vec2]  # Line E
print [vec1[i]*vec2[i] for i in range(len(vec1))]  # Line F
```

What is the output?

2. Consider the following Python program.

```python
def f(v1, v2 = 'Happy Thanksgiving!'):
  return [v1, v2]
print f(5)  # Line A
print f(v2 = 'Merry Christmas!', v1 = 3)  # Line B
print f(v2 = 3)  # Line C
print f(9, 10)  # Line D
print f(v1 = 20)  # Line E
print f(v3 = 7)  # Line F
```

What is the output? If a line causes an error, write ERROR as the output of that line. Also, explain why the error is produced.
3. Consider the following Python program.

```python
n = 10
def f1 ():
    m = 3
def f2 ():
    n = 4
    print m + n
    print m

def f3 ():
    print m + n
f2 ()
f3 ()
f1 ()
print 'Done!'
```

a) What is the output?
b) For each function, list each of its local variables and the line number(s) in which the local variables appear.
c) List all of the global variables that appear in the program and the line numbers in which they appear.

4. Suppose you know the following portion of the ASCII table.

- 'A' is 65.
- 'B' is 66.
- 'a' is 97.
- 'b' is 98.

Now, write out the binary representation for the word 'Apple'. Do this calculation based on what you know about the ASCII table and what’s given above.

5. Consider the following Python program.

```python
myDict = {}
myList = [1,2,3,4,5]
myStr = 'abcde'
cnt = 0

for c in myStr:
    myDict[c]=sum(myList[cnt:])
    cnt += 1
print myDict['b']
# Line A
print cnt
# Line B
print "Testing: ", myDict.items()  # Line C

for key,val in myDict.items():
    if val%2 == 0:
        myDict[key] = val/2
print myDict['b']
# Line D
print "Testing2: ", myDict.items()  # Line E
```
a) What output does Line A produce?
b) What output does Line B produce?
c) What output does Line C produce?
d) What output does Line D produce?
e) What output does Line E produce?

6. Convert the decimal number 186 to its binary and hexadecimal representations.

7. Convert the hexadecimal value 8FF3A0 to binary and decimal numbers.

8. Write a program that asks a user to enter a binary number (or string). Now, tell the user whether their binary number is even or odd.

a) One way to solve this problem is to convert the binary number to decimal, and then determine if the decimal number is even or odd.

b) However, for this problem, do not convert the binary number to decimal. By analyzing the binary string, you should be able to determine whether the number it represents is even or odd.

9. There are 36 possible combinations of two dice (die 1 and die 2). Write a program that uses a dictionary to create a table that produces the sum of two dice and all of the possible combinations that are possible for that sum. Below is an example of the table that your program should create.

<table>
<thead>
<tr>
<th>Total</th>
<th>Frequency (%)</th>
<th>Combinations</th>
</tr>
</thead>
<tbody>
<tr>
<td>2</td>
<td>2.8</td>
<td>[(1, 1)]</td>
</tr>
<tr>
<td>3</td>
<td>5.6</td>
<td>[(1, 2), (2, 1)]</td>
</tr>
<tr>
<td>4</td>
<td>8.3</td>
<td>[(1, 3), (2, 2), (3, 1)]</td>
</tr>
<tr>
<td>5</td>
<td>11.1</td>
<td>[(1, 4), (2, 3), (3, 2), (4, 1)]</td>
</tr>
<tr>
<td>6</td>
<td>13.9</td>
<td>[(1, 5), (2, 4), (3, 3), (4, 2), (5, 1)]</td>
</tr>
<tr>
<td>7</td>
<td>16.7</td>
<td>[(1, 6), (2, 5), (3, 4), (4, 3), (5, 2), (6, 1)]</td>
</tr>
<tr>
<td>8</td>
<td>13.9</td>
<td>[(2, 6), (3, 5), (4, 4), (5, 3), (6, 2)]</td>
</tr>
<tr>
<td>9</td>
<td>11.1</td>
<td>[(3, 6), (4, 5), (5, 4), (6, 3)]</td>
</tr>
<tr>
<td>10</td>
<td>8.3</td>
<td>[(4, 6), (5, 5), (6, 4)]</td>
</tr>
<tr>
<td>11</td>
<td>5.6</td>
<td>[(5, 6), (6, 5)]</td>
</tr>
<tr>
<td>12</td>
<td>2.8</td>
<td>[(6, 6)]</td>
</tr>
</tbody>
</table>

For example, for two dice that sum to 5, there is an 11.1% chance of this outcome occurring. Moreover, for a dice sum of 5, there are 4 different possibilities.

- die 1 = 1, die 2 = 4
- die 1 = 2, die 2 = 3
- die 1 = 3, die 2 = 2
- die 1 = 4, die 2 = 1

10. Redo Question #9. This time use the array data structure in numpy to create a table that produces the sum of two dice and all of the possible combinations that are possible for that sum.
11. Assume you have a list of words, where each word is of length 5. (Use the 5-letter-words.txt file from Lab #10.) Ask the user to enter a sequence of vowels (for example, a, e, and i). Now, write to the file output-words.txt all of the words that only contain the vowels a, e, and i in that order. Consonants can separate the vowels.

Here’s an example. First, ask the user for a sequence of vowels. Also, output the number of words that were printed to the output-words.txt file.

```
Please enter a string of vowels: aei
Number of words printed in output-words.txt: 3
```

Below, are the contents of the wordplay-output.txt file.

```
aegis
areic
naevi
```

You should try to write this program in the fewest number of lines possible (which must include comments and user-defined functions). A lot of code can be difficult to manage.

12. Write down the specifications of your home computer. How much memory do you have? How large is your hard drive? Do you have a 32 or 64-bit processor. Is your processor Intel, AMD, or something else? What operating system do you use? Now, compare your computer to the specifications of the machines in the lab?