The Final Exam from 2015. I tried to check every instance of a Python 2 idiom and change it to Python 3, but it is possible that I missed a print statement or two. In the last problem---the one about the clown---I took a somewhat different approach to building a new image array than what you saw in Assignment 6: Here you will see an all-zero array constructed, and then its values changed to represent the pixels in the transformed image. In our assignment this year we did not do the intermediate step of building an all-zero array.

1. Determine the value and type of each of the expressions below. For parts of this question that consist of two lines, assume that the statement in the first line is executed, and give the value and type of the second line. Note that it is possible for an expression to have type NoneType and value None, and also possible for an expression to be illegal and cause an error when it is evaluated.

(a) 9 // 4 * 4
(b) 1.0 * 5 / 10 == 5 // 10 * 1.0
(c) '46' + '12.3'
(d) s = "Boston College Eagles"
   (s[:1] + s[15:]).upper()
(e) y = [4, 3, 2, 1]
   y.append(0)
(f) y = (4, 3, 2, 1)
   y = y + (0)
(g) z = ((4, 5), (1, 2), (6, 7), (3, 1))
   z[0]
(h) z = ((4, 5), (1, 2), (6, 7), (3, 1))
   z[:1]
(i) d = {'head': (1, 2, 1), 'arm': (1, 2), 'hand': (1, 1, 2), 'leg': (1, 1, 1), 'foot': (1, 2, 1)}
   sum(d['head'])
(j) d={'head':(1,2,1), 'arm':(1,2), 'hand':(1,1,2), 'leg':(1,1,1), 'foot':(1,2,1)}
    [part for part in d if d[part][1] == 2]
2. Consider the following sequence of statements

```python
index=0
newstring=''
while index < len(s):
    if s[index] in 'aeiouAEIOU':
        newstring+='*
    else:
        newstring+=s[index]
index+=1
print (newstring.upper())
```

What is the value printed by this sequence if the initial value of `s` is

```
Murder most foul
```

3. What is the output printed when the following sequence of statements is executed?

```python
for m in range(2,40):
    if not (0 in [m%2,m%3,m%5]):
        print (m)
```

4. The following function takes an object of any sequence type (e.g., string, list, tuple) as an argument.

```python
def what3(s):
    return [s[j:] for j in range(len(s))]
```

(a) What is the type of the return value of `what3` if the type of `s` is string?

(b) What is the value returned by `what3('Python')`?

(c) What would be a better name for `what3`?
5. This problem is in two parts. In the first part you are to write a function that takes a list of words and returns a dictionary. In the second part, you will use this dictionary to answer some queries. The parts are independent of one another, so you can do part (b) even if you did not complete (a).

(a) Write a function that takes as a parameter a list of words, which you may assume to all be in lower case (e.g., ['aardvark','abate','academic',...]) and returns a dictionary in which each key is a word of the list, and the corresponding value is a list of the letters that appear in the word. For instance, with the word list above, the dictionary returned by the function would include items

'aardvark':['a','d','k','r','v']
'abate':['a','b','e','t']
'academic':['a','c','d','e','i','m']

(b) Suppose that d is a dictionary as above. Write a statement or sequence of statements that prints all the words in the dictionary that contain none of the letters 'aeiou'. (For example, if I do this with the Scrabble word list used in an earlier homework assignment, the output is rather long, and includes words like 'why', 'try' and 'brr'.)

6. This problem is also in two parts—again, you will write a function in the first part, and in the second, call the function to solve a problem.

(a) Write a function dice(n) that simulates n rolls of a pair of dice and returns a list of the outcomes, each of which is a tuple. For instance, dice(4) might return

[(1,3),(5,2),(4,6),(3,3)]

You may assume that import random has been executed somewhere before the function is called.

(b) Write an expression that contains a call to the dice function of (a) with an argument of 1000. The value of this expression should the number of times in 1000 rolls that the result was doubles (for example, (2,2)). In the example in (a), this expression will have the value 1. You will receive more credit for this problem if you solve it with a single line, using list comprehension, rather than with a sequence of statements leading to an expression with the required value.
7. Consider the following function:

```python
def what4(s):
    if len(s)==0:
        return []
    else:
        return what4(s[:len(s)-1])+[s]
```

(a) What is the value returned by what3(' ')?

(b) What is the value returned by what3('P')?

(c) What is the value returned by what3('Python')?

(d) What would be a better name for what3?
8. Shown below is the solution to the Eight Queens puzzle. The solution has been modified by the addition of a single statement at the beginning to print out the position in the function argument each time that the function is called. What are the first six lines printed when you call `eightqueens3(6,())`?

```python
def eightqueens3(size, position):
    print(position)
    if size > len(position):
        for j in range(size):
            if safe(position, j):
                newposition = position + (j,)
                result = eightqueens3(size, newposition)
                if result != None:
                    return result
            else:
                return position
```

(Just to remind you, the diagram below illustrates how positions are encoded by tuples. The picture on the left represents an unsafe position and would actually never be generated by the code shown above.)
The recent closing of the circus has shattered our clown's world and turned it upside down. Shown below are code for four image-transformation effects and the results of applying these to the clown image, along with original image. Match the code (a-d) to the corresponding image (I-IV). {HINT: Note that (a),(b),(d) all start the same, but (c) is different. Try to figure out (b) and (c) first and then use a process of elimination.)

```python
def a(im):
    height = len(im)
    width = len(im[0])
    newim = []
    for row in range(height):
        newcol = []
        for col in range(width):
            newcol.append([0,0,0])
        newim.append(newcol)
    for row in range(height):
        for col in range(width):
            newim[row][col] = im[(row+200)%height][(col+200)%width]
    return newim

def b(im):
    height = len(im)
    width = len(im[0])
    newim = []
    for row in range(height):
        newcol = []
        for col in range(width):
            newcol.append([0,0,0])
        newim.append(newcol)
    for row in range(height):
        for col in range(width):
            newim[row][col] = im[height-row-1][width-col-1]
    return newim

def c(im):
    height = len(im)
    width = len(im[0])
    for row in range(height):
        for col in range(width):
            im[row][col] = im[height-row-1][width-col-1]
    return im

def d(im):
    height = len(im)
    width = len(im[0])
    newim = []
    for row in range(height):
        newcol = []
        for col in range(width):
            newcol.append([0,0,0])
        newim.append(newcol)
    for row in range(height):
        for col in range(width):
            newim[row][col] = im[col%height][row%width]
    return newim
```