For the following questions, use these variable definitions.
\[
\begin{align*}
a &= 45 \\
b &= 4 \\
c &= 3.9999 \\
d &= "7"
\end{align*}
\]

What is the value and type of each of the following expressions or, if it won't compile, circle that answer

<table>
<thead>
<tr>
<th>type</th>
<th>value</th>
<th>circle if will not compile</th>
</tr>
</thead>
<tbody>
<tr>
<td>1. (2 pts)</td>
<td>(a \div b)</td>
<td>___________ ___________</td>
</tr>
<tr>
<td>2. (2 pts)</td>
<td>(a - 16 % b)</td>
<td>___________ ___________</td>
</tr>
<tr>
<td>3. (2 pts)</td>
<td>(\text{str}(b + 2) \times c)</td>
<td>___________ ___________</td>
</tr>
<tr>
<td>4. (2 pts)</td>
<td>(\text{int}(\text{str}(c)))</td>
<td>___________ ___________</td>
</tr>
<tr>
<td>5. (2 pts)</td>
<td>3 if 4 == 5 else 6</td>
<td>___________ ___________</td>
</tr>
<tr>
<td>6. (2 pts)</td>
<td>(\text{bin}(3))</td>
<td>___________ ___________</td>
</tr>
<tr>
<td>7. (2 pts)</td>
<td>(\text{str}(\text{str}(b)))</td>
<td>___________ ___________</td>
</tr>
<tr>
<td>8. (2 pts)</td>
<td>(b - 1 ^* 2)</td>
<td>___________ ___________</td>
</tr>
<tr>
<td>9. (2 pts)</td>
<td>(\text{not} 3 &lt; 4 \leq 4)</td>
<td>___________ ___________</td>
</tr>
<tr>
<td>10. (2 pts)</td>
<td>(\text{len}(d) + \text{len}('c'))</td>
<td>___________ ___________</td>
</tr>
</tbody>
</table>
11. (5 pts) Given:

```python
p = '''raw_input( "Enter a digit: " )'''
i = int( raw_input( "How many words are there? " ) )
print p + ' ( ' + str( i ) + ' )'''
```

Show what will be on the screen after these statements are executed and the user types 4 and then hits ENTER.

If there will be a runtime error, state that here.

(12) A programmer has been hired by a company to write a function that determines if a user's password is too short. Passwords are too short if they are shorter than a certain minimum length set by the company.

Note that this question is NOT about branching.

Do NOT write this function.

Do NOT write any code for this question.

ONLY answer this question:

12. (9 pts) What are the pieces of information this programmer needs to solve this problem?
   For each, state the source of information and how each would be implemented in python.
   (If there are no needed pieces of information, just state none.)
   Consider both good design as well as the fact that the company has a very specific minimum length.

<table>
<thead>
<tr>
<th>INFORMATION NEEDED</th>
<th>SOURCE</th>
<th>HOW TO IMPLEMENT</th>
</tr>
</thead>
</table>

13. (5 pts) A random number in [3, 4, 5, 6, 7] is chosen and stored. A different function is called when the chosen number is even or odd. Which problem form is this?

14. (3 pts) An if statement is a test. (circle your answer)  YES  NO
15. PART ONE (10 pts)

Draw arrows to show which values the parameter \( p \) refers to after lines 5 and 7 and which value the local variable \( x \) refers to after lines 8 when the following code is executed. If you need to show more numbers or change the values in RAM, do so. Assume that \texttt{callThee} is called first.

This code does compile and run correctly when \texttt{callThee} is called

```python
def callThee( ):  
    ''' … '''
0.     print 'THEE'
1.     x = 6
2.     p = callMe( x )
3.     print x
4.     return p

5. def callMe( p ):  
    ''' … '''
6.     print 'ME'
7.     p = 4
8.     x = p - 2
9.     return 5
```

Write the values that will be in RAM:

\[
\begin{array}{cccc}
  & & & \\
   & & & \\
   & & & \\
   & & & \\
   & & & \\
   & & & \\
   & & & \\
   & & & \\
   & & & \\
   & & & \\
\end{array}
\]

PART TWO (10 pts)

What is the output when \texttt{callThee} is called like this:

```python
def main():
    print callThee( )
```

\[
\begin{array}{cccc}
  & & & \\
   & & & \\
   & & & \\
   & & & \\
   & & & \\
   & & & \\
   & & & \\
   & & & \\
   & & & \\
   & & & \\
\end{array}
\]
16. (5 pts) When you want to use a function written by someone else, **what must you read** in order to know the details of how it should be used?

   a) the `def` line (the first line of the function definition)
   b) the docstring
   c) the body of the function
   d) the call of the function
   e) all of the above
   f) a) and b) only
   g) none of the above – because the caller should not need to know this

17. (5 pts) Circle all that is wrong in this code:

```python
def printName( name2 ):
    ''' Prints a person's name with "Mr. or Mrs." '''
    print( "Mr. or Mrs. " + name2 )
```

GO ON TO THE NEXT PAGE
18. (20 pts) A module whose file is named toasty.py contains a function named oblatt.

Here is a SnakePit session showing the help for that function:

oblatt(...)  
oblatt(object, object, object=1) -> str

The first two parameters must be strings, third must be int
Returns the str made of the first and second parameters each repeated
the third parameter number of times and then concatenated.
E.g.: the call with "Hi", "Lol", 4 yields the string: "HiHiHiHiLolLolLolLol"

Obviously you cannot define this function because it is in a module someone else created.
Your job is to complete the following function by writing the needed parameters and python statements that
do what the docstring for drawNamesBox says it will do.
Assume that from __future__ import print_function was written at the top of the file this is in.
Even though it's not the best, your function MUST use the oblatt builtin function
You do not need to show any testing for your function.
Do NOT write a main function (obviously).
Do NOT write a complete program (obviously).
ONLY complete the definition for drawNamesBox  (don't forget the parameters!)

def drawNamesBox(                           ):
    ''' Given two names, draws a three line pattern on the screen
    made of the names jammed together in a nice way.
    Here is an example of the output when the names are "Xavier" and "Suzy":
    SuzySuzySuzyXavierXavierXavierXavier
    >>>>>>>>>>>>>>>>>>>XavierSuzy
    XavierXavierXavierSuzySuzySuzy
    Notice that the first and last lines have both names three times.
    Notice that the middle lines have enough '>' characters
    to align all three lines at the right.
    '''
19. (35 pts) Randy Boxes

Define a module containing only one function named `randOrX` that returns a single random digit or the letter 'X' when the random digit is 4.

Define a module named `randyLine` which will contain these two functions:
--- a function that returns a string whose length is passed in where all the characters are the same random digit or an 'X' instead of 4.
--- a function that returns a string of length three where the three characters are each randomly a digit or an 'X' instead of 4.

Define a module which will contain functions that produce boxes that are always three lines high made of random digits, again, with 'X' instead of 4. This module must have these functions:
--- a function that produces a solid, three high box all lines use the same character all 3 lines in the box are the same length: randomly determined between 5 and 25 digits (inclusively).
--- a function that produces a 3 x 3 square, all of the 9 characters might be the same or different (random).

Write a `main` function to test the functions in your module by making only two function calls.

Start your `main` function and each module's code on a left page in your blue book.
Clearly mark the filename at the top of the pages with modules.

Do not forget that you must always think about procedural problem solving.

We are showing two runs of your program because of the random nature of the digits and length being used.

Here is a sample run:

<table>
<thead>
<tr>
<th>287</th>
<th>538</th>
<th>776</th>
</tr>
</thead>
<tbody>
<tr>
<td>3333333333333333</td>
<td>3333333333333333</td>
<td>3333333333333333</td>
</tr>
</tbody>
</table>

Here is another sample run:

<table>
<thead>
<tr>
<th>982</th>
<th>110</th>
<th>X2X</th>
</tr>
</thead>
<tbody>
<tr>
<td>5555555555</td>
<td>5555555555</td>
<td>5555555555</td>
</tr>
</tbody>
</table>

Here is another sample run:

<table>
<thead>
<tr>
<th>555</th>
<th>60X</th>
<th>321</th>
</tr>
</thead>
<tbody>
<tr>
<td>XXXXXXXX</td>
<td>XXXXXXXX</td>
<td>XXXXXXXX</td>
</tr>
</tbody>
</table>

Recall that 'digit' means a number between 0 and 9, inclusive.

You do not have to write any docstrings as long as you choose good names for variables, parameters, module level and global constants and functions.

Start your `main` function and each module's code on a left page in your blue book.
Write your code across both pages, moving from far towards the far right.
Clearly mark the module's filename at the top of the pages where they start (each on a left hand page).