1. Vocabulary Matching: (15 points)
Write the number from the correct definition in the blank next to each term on the left:

<table>
<thead>
<tr>
<th>Term</th>
<th>Definition</th>
</tr>
</thead>
<tbody>
<tr>
<td>compound data type</td>
<td>1. A part of a string specified by a range of indices.</td>
</tr>
<tr>
<td>slice</td>
<td>2. A collection of key-value pairs that maps from keys to values.</td>
</tr>
<tr>
<td>traverse</td>
<td>3. A compound data type whose elements can be assigned new values.</td>
</tr>
<tr>
<td>mutable</td>
<td>4. The process of calling the function that is currently executing.</td>
</tr>
<tr>
<td>increment</td>
<td>5. Repeated execution of a set of statements using either a recursive function call or a loop.</td>
</tr>
<tr>
<td>decrement</td>
<td>6. Multiple variables that contain references to the same object.</td>
</tr>
<tr>
<td>element</td>
<td>7. To iterate through the elements of a set, performing a similar operation on each.</td>
</tr>
<tr>
<td>aliases</td>
<td>8. Any of the data types that consist of an ordered set of elements, with each element identified by an index.</td>
</tr>
<tr>
<td>sequence</td>
<td>9. A list that is an element of another list.</td>
</tr>
<tr>
<td>nested list</td>
<td>10. To increase the value of a variable by one.</td>
</tr>
<tr>
<td>clone</td>
<td>11. To decrease the value of a variable by one.</td>
</tr>
<tr>
<td>immutable type</td>
<td>12. One of the values in a list (or other sequence). The bracket operator selects an __________ of a list.</td>
</tr>
<tr>
<td>dictionary</td>
<td>13. A type in which the elements cannot be modified. Assignments to elements or slices of these types cause an error.</td>
</tr>
<tr>
<td>recursion</td>
<td>14. To create a new object that has the same value as an existing object.</td>
</tr>
<tr>
<td>iteration</td>
<td>15. A data type in which the values are made up of components, or elements, that are themselves values.</td>
</tr>
</tbody>
</table>
2. Write Code (15 points)
Write a function \texttt{return\_smallest} that accepts 3 parameters (x,y,z) and returns the smallest of the three. For example, \texttt{return\_smallest(7, -34, 23.8)} should return -34. Make sure that your function works for test cases such as \texttt{return\_smallest(5,5,4)}.

3. Program Comprehension (3 points)
\begin{verbatim}
def n_lines(n):
  print "Line!"
  if n >= 0:
    n_lines(n-1)
\end{verbatim}

How many times will the string “Line!” be printed when \texttt{n\_lines} is called with \texttt{n=4}?
Number__________

4. Write Code (2 points)
Write a function with infinite recursion named \texttt{run\_forever}. Your function should have no parameters, and it should run forever when called (on an ideal computer, in a real computer it would eventually run out of memory.) You may add a print statement if you wish.
5. Robot Directions (10 points)
The following code makes the robot drive the trajectory drawn in the box to the right.

```python
def turn90degrees():
    turnRight(1, 1)

def nudge(x):
    forward(1, x)

nudge(1)
turn90degrees()
nudge(1)
nudge(2)
```

Draw the robot's trajectory when the following code is executed. Start the robot in the middle of the box and use arrow heads (as above) to indicate each movement.

```python
def turn90degrees():
    turnRight(1, 1)

def nudge(x):
    forward(1, x)

nums = [1, 2, 3, 4]

for i in nums:
    if (i % 2 == 0):
        turn90degrees()
nudge(i)
```

6. Fill in the blank (2 points)
In python, the `=` operator performs _____________________ while the `==` operator performs ____________________.
7. Python Expression Evaluation (20 points)
For this question, assume the following statements have already been entered and interpreted:

```python
a = [ True, 7, ["Cherry", "Apple","Plum"], 56, [4, 5, 6], 84 ]
b = a
c = a[0:4]
d = a[2]
d[2] = "Peach"
```

Pretend that you are the Python Interpreter (IDLE window). What do you print or return when each of the following statements are entered?

Example:  
```
a[0] 
```

Result: **True**

Example:  
```
a[4:6] 
```

Result: _ [ [4,5,6], 84 ]_

1.  
```
a[4][0] 
```

Result: _____________________________

2.  
```
d 
```

Result: _____________________________

3.  
```
c 
```

Result: _____________________________

4.  
```
a[2][2] 
```

Result: _____________________________

5.  
```
b[:2] 
```

Result: _____________________________

6.  
```
b[-2] 
```

Result: _____________________________

7.  
```
c[-2] 
```

Result: _____________________________

8.  
```
print "Pumpkin %.3f" %3.1459 
```

Result: _____________________________

9.  
```
(5 > 10) or (5 > 3) 
```

Result: _____________________________

10.  
```
34 % 10 
```

Result: _____________________________
8. Write Code (10 points)
Write a function `changeLetter(aString, index, newLetter)` that will replace the letter stored at index in `aString` with the contents of `newLetter` and return the new string without modifying the original string! For example, `changeLetter("Python is great!", 10, "G")` will return the string "Python is Great!"

9. Write Code (5 points)
Write a function `changeValue(aList, index, newValue)` that will replace the element stored at index in `aList` with the contents of `newValue`. It should NOT return the list. For example after the following commands:

```python
a = [5, True,"Test",10]
changeValue( a, 3, "Hi!"")
```

The list `a` will be `[5, True,"Test","Hi!"]`
10. Write Code! (20 points)

Write a function called roboFlute that takes no parameters. The roboFlute function will watch each of the robot's 3 light sensors (values obtained using the getLight("loc") function and specifying a location out of the set ("left" / "right" / "center") ) and play a beep that is ½ second long if a light sensor is covered. You know a light sensor is covered when the value it returns is larger than 1000. Each of the three sensors should play a different note, as follows: “left” = 800Hz, “center” = 440Hz, “right” = 220Hz. The robotFlute function should perform the above actions for 25 seconds and then return.

API Hints: beep(time_in_seconds, frequency_in_Hz), value = getLight("location")
11. Write Code! (15 points)

Write a function `reverseList(aList)` that will return a reversed copy of aList. For example, after the following:

```
a = [ 5, 10, True, "Hi!"
```

```
b = reverseList(a)
```

The list `b = ["Hi!", True, 10, 5]`, while `a = [5, 10, True, "Hi!"]`.

12. Write Code: (10 Points)

Write a function `findJ(aString)` that uses a `while` loop to find the index of the first occurrence of the letter 'J' in the `aString` parameter. The function should return the index it found. For example, `findJ("This is Jays String")` should return the number 8. Note that you should find both UPPERCASE J's and lowercase j's! If you do not find a J or a j you should return -1.
Extra Credit (1 point each)

What is the one sensor on the Scribbler that does not detect light of one form or another?
_____________________

What is the decimal representation of the binary number \{ 101101\}? __________

What is the hexadecimal representation of the decimal number 34? __________

What does CSS stand for? C__________ S________________ S_______________