ECE-203 Programming for Engineers
Laboratory Experiment 1

Name: ___________________________________

Python makes heavy usage of **lists** and **tuples**. These were briefly introduced in the first lecture and will be covered in more depth next week. This week you will gain some hands on experience with Python operating in interactive mode by experimenting with these fundamental data types.

Start up **python** in interactive mode. We are going to do some basic experimenting with the list and tuple types. Start by typing in the following:

```
>>> A = [3, 4, 5, 6]
>>> B = [10, 20, 30, 40]
>>> D = (98, 99, 100, 101, 102)
```

(1 Points) What does the following command produce?

```
>>> A
```

(1 Points) Now, what does the following command produce?

```
>>> A[2]
```

(1 Points) Now, what does the following set of commands produce?

```
>>> A[2] = B
>>> A
```

(1 Points) Now, what does the following command produce?

```
>>> A[2]
```

(1 Points) Now, what does the following command produce?

```
>>> A[2][3]
```

(1 Points) Now, what does the following command produce? Why?
Does this give you a clue about how the `[ ]` operator works?

```
>>> A[1][3]
```

(1 Points) Now, what does the following command produce?

```
>>> D[1]
```

(1 Points) Now, what does the following command produce? Why?

```
>>> D[1] = 33
```

(1 Points) Now, what does the following command produce?

```
>>> A[1:4]
```

(1 Points) Now, what does the following command produce?

```
>>> D[0:3]
```

(1 Points) Now, what does the following command produce?

```
>>> D[0:4:2]
```

(1 Points) Now, what does the following command produce?

```
>>> type(D)
```

(1 Points) Now, what does the following command produce?

```
>>> type(D[1:3])
```

(1 Points) Now, what does the following command produce?

```
>>> type(D[1])
```

(1 Points) Now, what does the following command produce?

```
>>> type(A[2])
```

(1 Points) Now, what does the following command produce?

```
>>> A[3] = D
>>> A
```

(1 Points) Now, what does the following command produce?

```
>>> type(A[3])
```

(1 Points) We will introduce `for` loops and iterators formally in lecture, but let's play with one for now – just to get a feel for it. What does the following series of commands produce? Remember, in Python indentation counts!
(Note: In interactive mode, hitting enter on an empty line ends a block.)

```
>>> for i in B:
...     print i+1
...
```

(2 Points) Finally, what does the following set of commands produce?

```
>>> foo = "The tuple contains: %d,  %d,  %d, %d, and %d" % D
>>> print foo
>>> foo[4:18]
```