ENGIE 1006
Name: ______________________
UNI: ______________________
Fall 2016
Exam 3
December 12th, 2016
Time Limit: 70 Minutes

My signature on this line is required. My signature indicates that I understand that I am not permitted to retain any written record, photograph, or facsimile of this test material. I am not permitted to communicate the substance of this test material to other students. My signature on this line is *required* or this test is void and shall not be graded. **I hereby sign here:**

This exam contains 9 pages (including this cover page) and 3 questions. Total of points is 100.

Grade Table (for teacher use only)

Question	Points	Score
Multiple Choice	30	
Short Answers	30	
Writing Codes	40	
Total:	100	

1 Multiple Choice (30 pts)

For each question, mark your selection unambiguously. On some questions, you will be asked to **circle ALL that apply**. In that case, the answer may be one or more of the choices presented. You must **clearly delineate** which choices you selected; we will not give you the benefit of the doubt in ambiguous cases. You may write the letter(s) of your final selection next to question to be safe, and we will treat that as your final answer.

(1) Read the following code snippet:

```
d = { "lunch": "ciabatta sandwich",
        "dinner": "spicy tuna roll" }
set1 = set()
for key in d:
    set1.add(key)
meals = ["dinner", "breakfast", "brunch", "lunch", "dinner", "late night snack"]
set2 = set()
for meal in meals:
    if meal in d:
        set2.add(meal)
```

Which of the following evaluates to True? Select ONE.

A. set1 == set2 and set1 is set2

B. set2 == set(meals)

C. **set1.union(set2) == set1.intersection(set2)**

(2) What will the following code snippet print out? Select ONE.

```
def foo(n):
    if (n % 2 == 0 and n % 3 == 0):
        b = 6
    if (n % 3 == 0):
        b = 3
    if (n % 2 == 0):
        b = 2
    return b
print(foo(36))
print(foo(5))
```

A.
```
6
5
```

B.
```
6
UnboundLocalError: local variable 'b' referenced before assignment
```

C.
```
6
3
2
5
```

D. **2**
UnboundLocalError: local variable ’b’ referenced before assignment

E.
```
2
5
```

(3) Consider the following code snippet:
```
d = {"Mariah": "Carey"}
d[(1,2)] = 3
d[1] =  "1"
d[3] = "1,2"
```
Which of the following statements will return False? (Circle ALL that apply):

A. "Mariah" in d

B. **"1" in d**

C. (1,2) in d

D. **d[3] == (1,2)**

(4) Consider the following code snippet:
```
import numpy as np
arr = np.array([[1, 1, 1], [2, 2, 2]])
```
Which of the following statements will evaluate to True? (Circle ALL that apply)

A. arr * 2 == np.array([[1, 1, 1], [2, 2, 2], [1, 1, 1], [2, 2, 2]])

B. arr / 2 is np.array([[0.5, 0.5, 0.5], [1, 1, 1]])

C. **arr.shape == (2,3)**

D. arr.tolist() == [1, 1, 1, 2, 2, 2]

(5) Select the print value of this code snippet. Circle ONE.
```
import numpy
my_array = numpy.array([None, 0, "", []], dtype=bool)
print(numpy.any(my_array))
```
A. True

B. **False**

(6) Read the following code snippet:
```
a = {18, 2, 9, 8}
b = [8, 9, 2, 18]
c = {
    "sad": "machine",
    "lion": "heart",
    "shel": "ter"
}
```
Which of the following statements would throw an error? (Circle ALL that apply)

A. **a.sort()**

B. b.sort()

C. **c.sort()**

D. sorted(a)

E. sorted(b)

F. sorted(c)

(7)
```
a = [1, 2, 3]
b = [0, 0, 0]
count = 1
j = 0
while (j < len(b)):
    b[j] = count
    count += 1
    j += 1
c = a
```
Which of the following returns True? (Circle ALL that apply)

A. a is b

B. not (a is c)

C. not (a == b)

D. **b == c**

(8) What is printed by the following code snippet? Circle ONE.
```
def f(n):
    if n == 1:
        return 1
    return f(n-1) + f(n-2)

print(f(5))
```
A. 5

B. 13

C. 8

D. **This would result in an error.**

2 Short Answers (30 pts)

(1)
```
def mystery_func(s):
    if(s == ""):
        return True
    return (s[0] == s[-1]) and mystery_func(s[1:-1])
```
Give a very brief description of what mystery_func does, and provide an example of a non-empty value of s for which mystery_func will return True.

ANS: It checks if a string is a palindrome. Example input: any palindrome

(2) Consider this code snippet:
```
count1 = [1, 2, 4, 6, 7]
for val in count1:
    val = str(val)

count2 = [1, 2, 4, 6, 7]
i = 0
while i < len(count2):
    val = str(count2[i])
    i += 1
```

After running this code, what will be the type of count1[0] and count2[0]? Clearly write your answers in the space below.

ANS: They will both still be integers

(3) Below is the skeleton for a recursive function that takes in two non-negative integers, m and n, and returns their product. Fill in the blanks: (You may NOT use the * or / or % operators): **ANS:**
```
def mult(m, n):

    if (n == 0):

        return 0

    else:

        return m + mult(m, n-1)
```

(4) Consider the following code snippet:

```
names = ["Padfoot", "Moony", "Prongs"]
history_of_magic = ["100", "98", "95"]
defence_against_dark_arts = ["94", "100", "97"]

d = {}
for i, name in enumerate(names):
    d[name] = dict(mark1=defence_against_dark_arts[i], mark2=history_of_magic[i])

for name in names:
    print(name, d[name])
```

In the space below, write the printed output of this code. Note that contents of a dictionary may be printed in a non-deterministic order; you may write in any valid ordering of the contents.

ANS:
Padfoot {'mark2': '100', 'mark1': '94'}
Moony {'mark2': '98', 'mark1': '100'}
Prongs {'mark2': '95', 'mark1': '97'}

(5) Consider the following code snippet:

```
x = \
    [
        {
            0: ["1006", "4701"]
        },
        {1},
        ("4701",)
    ]
```

Write out the print value of the following statements in the corresponding spaces provided. If the statement would result in an error, simply write "ERROR".

```
print(x[0][0][0])                  1006

print(x[1][0])                     ERROR

print(x[0][0][1] == x[2])          False

print(x[0][0][1] == x[2][0][0])    False
```

3 Writing Codes (40 pts)

For the following functions, you may NOT use ANY for or while loops. **If your code contains a for or a while loop, you will receive a 0 for that question.**

Make concise, explanatory comments and name your variables with purpose. If your code does not work and it is difficult to parse the logic of your code, it will be equally difficult to award you partial credit, if at all.

(1) **Blast off: (15 points)**
Write a recursive function countdown(n,m) that takes two positive integers, n and m. It starts by printing n and then "counts down" n by a decrement of m, printing the decremented value. It continues to do this until it reaches 0 or a negative number, at which point, **instead** of printing 0 or a negative number, it prints "Blast off!".
Sample usage:

```
>> blast_off(5, 1)
5
4
3
2
1
Blast off!

>> blast_off(7, 4)
7
3
Blast off!
```

```
-----------------------------------------------------
def count_down(n, m):
    if n<=0:
        print("Blast off!")
    else:
        print(n)
        count_down(n-m,m)
```

(2) **Left-Truncatable Prime (25 points)**
In number theory, a left-truncatable prime is a prime number for which, if the leading ("left") digit is successively removed, all resulting numbers will be prime as well. For example, 9137 is a left-truncatable prime, since 9137, 137, 37 and 7 are all prime.
Assume you have a function available to you, **is_prime(m)**, which takes an integer m and returns True if m is prime, and False if it is not.

Write a recursive function is_left_truncatable_prime(n), that returns True if n is a left-truncatable prime, and False otherwise.

```
def is_left_truncatable_prime(n):
    if n//10 == 0:
        return is_prime(n)
    else:
        return is_prime(n) and is_left_truncatable_prime(int(str(n)[1:]))
```

Left blank in case you need extra space.