There are two sections, each worth 50 points. One of the questions in section 2 has an extra credit component worth up to 5 points. Partial credit is possible for each question. The maximum score for the test is 100 points (or 105 including the extra credit).

It is essential that you **PUT YOUR NAME ON ALL TEST MATERIALS**. It can be difficult to identify the author of an unsigned test and it would be better to avoid this problem.

**Section 1**: Below you will find several Pieces of Code followed by a question and a place to fill in an answer. Assume that there are no bugs in the code that will make the system crash, although some code may not solve a particular problem perfectly. If you find anything that you think is a bug, there is either a typo (and I should fix it for everyone) or you are mistaken.

**Sample Question A**

```python
def replace_vowels_with_asterisk(string):
    output = ''
    last_is_vowel = False
    for character in string:
        if (character in 'aeiouAEIOU') \
            or (character in 'yY' and not(last_is_vowel)):
            last_is_vowel = True
            output = output + ' * '  
        else:
            last_is_vowel = False
            output = output + character
    return(output)
```

output = replace_vowels_with_asterisk('two hundred twenty')

**Question 1**

Question: What is the value of output?

Answer:
def test_number_for_properties(number):
    import math
    extent = 0
    evenness = (number%2 == 0)
    week_multiple = (number%7 == 7)
    is_57 = (number == 57) or (math.ceil(number) == 57) or 
        (math.floor(number) == 57)
    is_integer = int(number) == number
    if number < 10:
        extent = 10
    elif number < 20:
        extent = 20
    elif number < 100:
        extent = 100
    else:
        extent = 101
    print(number,'is ',end='')
    if is_integer:
        print('an integer, ',end='')
        if evenness:
            print('an even number',end='')
        elif week_multiple:
            print('a multiple of 7 ',end='')
        else:
            print('an odd number ',end='')
    elif is_57:
        print('approximately equal to 57 ',end='')
    else:
        print('not an integer ',end='')
    if extent == 101:
        print('and is greater than 100. ')
    else:
        print('and is less than ',extent,'.',sep='')

test_number_for_properties(1)
test_number_for_properties(57.57)
test_number_for_properties(10)

Question: What will print out from the three function calls above?
Answer:
import turtle
my_screen = turtle.Screen()
turtle1 = turtle.Turtle()
turtle2 = turtle.Turtle()
turtle1.speed(0)
turtle2.speed(0)

def move_both_turtles(distance):
    turtle1.fd(distance)
    turtle2.fd(distance)

def turn_turtles_opposite(degrees):
    turtle1.left(degrees)
    turtle2.left(degrees+180)

def square(turtle,side):
    for rep in range(4):
        turtle.fd(side)
        turtle.left(90)

def turtle_square_eight():
    turtle1.pd()
    turtle2.pd()
    turn_turtles_opposite(45)
    square(turtle1,40)
    square(turtle2,40)
    turtle1.pu()
    turtle2.pu()

turtle_square_eight()

Question: Draw the shape (approximately) that the turtles would draw.
Answer:
Question 4:

def get_sequence_pairs_from_string(string):
    output = ''
    for character1 in string:
        for character2 in string:
            if character1 != character2:
                output = output + character1 + character2 + ' '
    return(output[:-1])

output = get_sequence_pairs_from_string('abc')

Question: What is the value of output?
Answer:
Section 2: Write Functions as specified.

Question 5: Write a program that produces a rectangle consisting of a combination of two characters. There should be four parameters: height (the number of lines), line_length (the length of each line), character1 (one character) and character2 (a different character). For each line of length line_length, generate a random number that is greater than 0 and less than line_length (use the random_number function below with line_length-1 as the maximum). The random number generated is the number of instances of character1 you should have on that line. The remaining characters on that line should be instances of character2.

```python
def random_number(maximum):
    import math
    import random
    return(math.ceil(random.random() * maximum))
```
Question 6: Write a program that produces a visual simulation of a drumbeat. The function takes two parameters: an instruction and a tempo. The tempo is the number of beats per minute. The instruction is a string of some combination of three characters: P, X and O, where P represents a pause, X represents an emphasized drumbeat and O represents an ordinary drumbeat.

Create an infinite while loop that repeats the pattern indicated by the instruction string, waiting one beat (using the time.sleep function) in between instructions. For example, given a tempo of 120 beats per minute, each beat will last 60/120 (or 0.5) seconds.

Each instruction should cause one line to print as follows:

- P (Pauses) should be printed as '—'

- X (Emphasis) should be printing as '***********Bum***********'

- O (A normal beat) should be printed as '*******Ba*******'

For example, the function call: `visual_drumbeat('OXPOOP',120)` should repeat the following printout, with a wait of .5 seconds before each line is printed.

```plaintext
*******Ba *******
*********** Bum ***********
---
******* Ba *******
******* Ba *******
---
*******Ba******
***********Bum***********
---
*******Ba******
***********Bum***********
---
```