Test 2 – Paper and Pencil part

Name: YANG ZHANG.

Honesty Pledge:

Included in this test is an ***Honesty Pledge*** that is exactly the same as the one which you should have read before the exam. Re-read the Honesty Pledge at the beginning of the exam.

When you are finished with this test, email your instructor saying either:

- I agree with what the Honesty Pledge says, OR
- I ***do NOT agree*** with what the Honesty Pledge says ***and will talk with you privately soon after the test.***

Have you:

- **Successfully completed** and committed ***all the programming exercises from Session 13?***
- Checked your ***paper-and-pencil* exercises from Session 13** against the answers online?

If not, DO NOT BEGIN THIS EXAM!
Instead, see your instructor to find out what to do.

Two parts (this is Part 1 – Paper-and-Pencil):

For this part, the ONLY external resource you may use is a single 8½ by 11-inch sheet of paper, with whatever you want on it, typed or handwritten or a combination of the two. You may use BOTH sides of the sheet. You must have prepared the sheet *before* beginning the exam.

Communication:

For both parts of the exam, ***you must not communicate with anyone*** except your instructor and his assistants, if any. In particular:

- You must not talk with anyone else or exchange information with them during the test.
- ***You must NOT use email,*** *chat* or the like during the test.

Time limit:

You have ***3 hours*** to complete the entire exam – its *paper part* and its *computer part*. Do the paper part first (using only your prepared 1-page-front-and-back sheet). Do not return to the paper part after you begin work on the computer part.

Problem	Points Possible	Points Earned	Comments
1	6		
2	8		
Total (of 100 on the test)	14		

1. (6 points) Consider the code snippet below. It is a contrived example with poor style, but it will run without errors. What does it print when *main* runs?

   Write your answer in the box to the right.

```
def main():
    numbers = [6, 50, 30, [3, 2, 1]]

    print('Before:')
    print_them(numbers)

    x = foo(numbers)
    numbers[3] = x

    print()
    print('After:')
    print_them(numbers)


def print_them(sequence):
    for k in range(len(sequence)):
        print(k, sequence[k])


def foo(sequence):
    sequence[1] = 999
    return 88
```

Output:

Before:

0 6
1 50
2 30
3 [3, 2, 1]

After:

0 6
1 999
2 30
3 88

2. (8 points) Consider the code in the box below. To the right of the box of code, draw the ***box-and-pointer diagram*** for what happens when ***main*** runs. In the space at the bottom, show what the code would ***print*** when ***main*** runs.

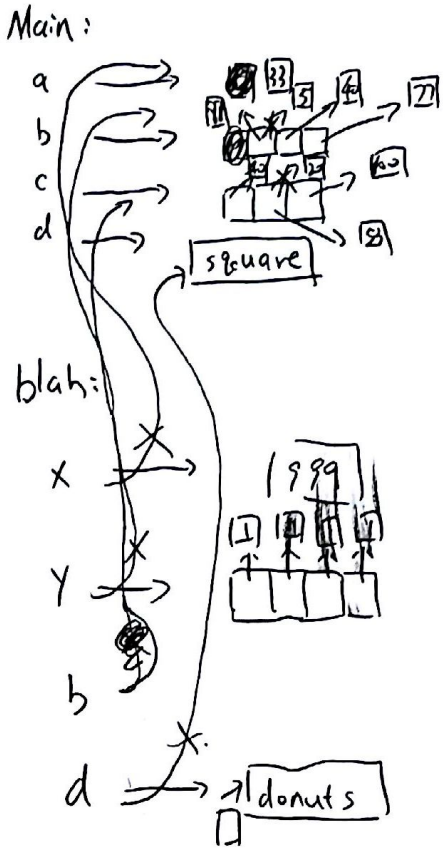
```
def main():
    a = 33
    b = [5, 40, 77]
    c = [40, 20, 100]
    d = 'square'
    blah(a, b, c, d)

    print('A.', a)
    print('B.', b)
    print('C.', c)
    print('D.', d)


def blah(x, y, b, d):
    x = 999
    y[0] = 88
    b[1] = 53
    d = 'donuts'
    y = [1, 1, 1, 1]
    y[2] = 66

    print('1.', x)
    print('2.', y)
    print('3.', b)
    print('4.', d)
```

Draw box-and-pointer diagram below here



What prints when ***main*** runs?

1. 999
2. [1, 1, 66, 1]
3. [40, 53, 100].
4. donuts.

A. 0
B. [88, 40, 77].
C. [40, 53, 100].
D. square.