

15-112 S22 Lecture3 Quiz 4

Name: _____

Andrew ID: _____

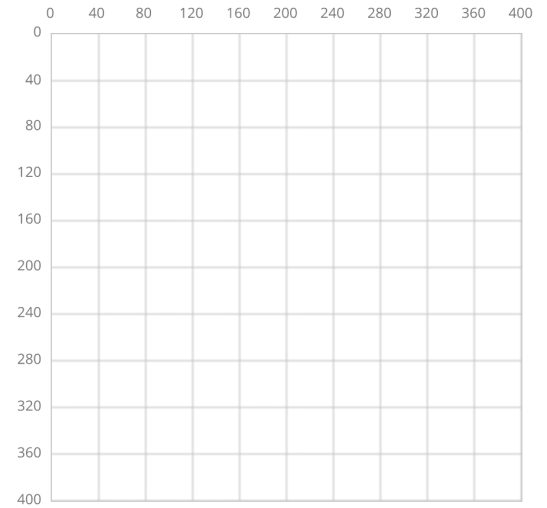
1. [20 pts] Graphics CT

Sketch what this will draw when run on a 400x400 canvas

```
from cmu_graphics import *
import math

def redrawAll(app):
    drawLine(200, 0, 200, 400)
    drawLine(0, 200, 400, 200)
    x, y = 300, 200
    t = 0
    for q in range(4):
        t += 90
        u, v, = x, y
        x = 200 + 100 * math.cos(math.radians(t))
        y = 200 - 100 * math.sin(math.radians(t))
        drawLine(u, v, x, y)

runApp()
```



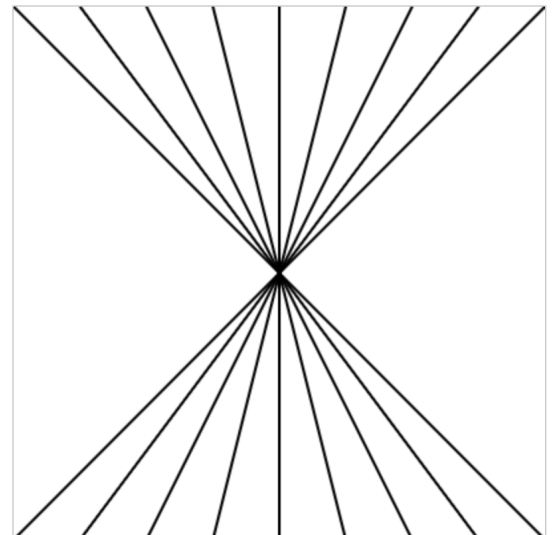
2. [10 pts] Graphics Fill in the Blank

Fill in the blank with one line of code so that the following code draws this image.

```
from cmu_graphics import *

def redrawAll(app):
    for x in range(0, app.width+1, 50):
        # _____

runApp()
```



3. [70 pts] Bookshelf Class

Write the Bookshelf class so the following test function passes (and do not hardcode anything, so any reasonably similar uses of the Bookshelf class would also work properly). We can add books to a shelf, or we can move them from one shelf to another.

Note: You may assume we will not try to move a book if it is not already on a shelf, BUT we will have to make sure it is moving to a shelf with enough space.

```
def testBookshelfClass():
    s1 = Bookshelf(5)
    assert(s1.books == []) #No books on the shelf
    assert(s1.getSpace() == 5) #There is space for 5 more books
    assert(s1.addBooks(['Math','Fantasy']) == True) #We can add books
    assert(s1.books == ['Math','Fantasy'])
    assert(s1.getSpace() == 3) #We can fit 3 more books

    assert(s1.addBooks(['Python']) == True) #True means we added successfully
    assert(s1.books == ['Math','Fantasy','Python'])
    assert(s1.getSpace() == 2) #Can fit 2 more.

    assert(s1.addBooks(['A','B','C','D','E']) == False) #False, not enough
    # space!
    assert(s1.books == ['Math','Fantasy','Python']) #No books added
    s2 = Bookshelf(2) #Make a tiny bookshelf for two books
    assert(s2.books == [])
    #Move 'Math' and 'Python' from s1 to s2
    assert(s1.moveBooks(s2, ['Math','Python']) == True)
    assert(s1.books == ['Fantasy'])
    assert(s2.books == ['Math','Python'])
    assert((s1.getSpace() == 4) and (s2.getSpace() == 0))

    #If there isn't enough space, we don't move anything and return False
    assert(s1.moveBooks(s2, ['Fantasy']) == False)
    assert(s1.books == ['Fantasy'])
    assert(s2.books == ['Math','Python'])
```