1. Write a function called `saveSensors` which takes an integer parameter, \( n \). The function should store \( n \) light sensor readings from the robot in a list. The function should return the list. Light sensor readings can be obtained by using the function `getLight("center")`.

2. Write a function which takes an integer, \( n \), as a parameter. It that will print the numbers 1,2,3...\( n \) in that order. You must use only a FOR loop. When you have that working, re-write the function to use only a WHILE loop.

3. What is printed when the following code is executed?

   ```python
   l = ["open", "close", "in", "out", "up", "down"]
   for i in range(0,6,2):
       print l[i]
   ```

4. Assume `turn90degrees()` has been defined as below so that the robot turns 90 degrees to the right. Assume `nudge(x)` has been defined to move the robot forward \( x \) units.

   ```python
   def turn90degrees():
       turnRight(1,1)

   def nudge(x):
       forward(1,x)
   ```

   The following code makes the robot draw the trajectory on the right.

   ```python
   nudge(1)
   turn90degrees()
   nudge(1)
   nudge(2)
   ```

   Draw the robot's trajectory when the following code is executed. Label the length of each move (nudge) using numbers as in the above example.

   ```python
   turns = [2,6]
   for index in [2,2,6,2,1]:
       if index in turns:
           turn90degrees()
           nudge(index + 1)
   ```
6. Examine the following code. Draw a sketch of what would be shown on-screen if the draw function were called.

```python
def draw():
    pic = takePicture()
    for x in range(getHeight(pic)):
        for y in range(0, getHeight(pic)):
            pix = getPixel(pic, x, y)
            if x == y:
                setRed(pix, 0)
                setGreen(pix, 0)
                setBlue(pix, 0)
    show(pic)
```

7. The code shown in problem #6 is inefficient. It examines every pixel in the image (all 49,000+ of them!) and only changes a few (how many specifically?) when the x coordinate is the same as the y coordinate. Rewrite this code to be more efficient. Why is your implementation better than the one given in problem #6?

8. Act like the python interpreter and "run" the following program. What does this program print?

```python
def fun1(x):
    print "Fun1 x:", x
    return x * 2

print "Start"
y = fun1(10)
if (5 > y ):
    print y
elif ( 15 > y ):
    print y + 10
elif (25 > y ):
    print y + 100
elif (35 > y):
    print y + 1000
else:
    print y + 10000

print "End"
```