

Your Name: \_\_\_\_\_ Your TA's Name: \_\_\_\_\_

1 / 9

## CS 1301 CS1 with Robots Fall 2008 – Exam 3

This test has ten (10) problems on Nine (9) pages. Some problems have multiple parts.

<b>Problem</b>	<b>Score</b>	<b>Possible Points</b>
1. Vocabulary		15
2. Fill in the Blank (1)		8
3. Fill in the Blank (2)		12
4. N Lines		4
5. Which walls?		8
6. Which Image?		8
7. A rubber spine		9
8. printFile()		12
9. maxValues()		15
10. color2Gray		18
<b>Total:</b>		109

Your Name: \_\_\_\_\_ Your TA's Name: \_\_\_\_\_

2 / 9

1. Vocabulary Matching (15 points). Write the number of the correct definition from the right column before each word in the left column.

___ Handle	1. Contains printable characters organized into lines separated by newline characters.
___ Text File	2. A named entity, usually stored on a hard drive, floppy disk, or CD-ROM, that contains a stream of characters.
___ Local Variables	3. % takes a format string and a tuple of expressions and yields a string that includes the expressions, formatted according to the format string.
___ Format Operator	4. A string that contains printable characters and format sequences that indicate how to format values.
___ Pickle	5. To write a data value in a file along with its type information so that it can be reconstituted later.
___ Short Circuit Evaluation	6. An error that occurs at runtime.
___ Lambda	7. To prevent an exception from terminating a program using the <code>try</code> and <code>except</code> statements.
___ Global Variables	8. Statement used to signal an exception.
___ Semantic errors	9. An error produced by Python when the user presses F5 to run the code.
___ File	10. Raised by the runtime system if something goes wrong while the program is running.
___ Runtime errors	11. Problems with a program that compiles and runs but doesn't do the right thing. Example: An expression may not be evaluated in the order you expect, yielding an unexpected result.
___ Exception	12. Can be seen through a program module, even inside of functions.
___ Raise	13. Names defined within a function, are only visible within that function.
___ Format String	14. A block of code which can be executed as if it were a function but without a name.
___ Syntax Errors	15. When a boolean expression is evaluated the evaluation starts at the left hand expression and proceeds to the right, stopping when it is no longer necessary to evaluate any further to determine the final outcome.

Your Name: \_\_\_\_\_ Your TA's Name: \_\_\_\_\_

3 / 9

## 2. Fill in the blanks 1 ( 8 points)

Python has several compound data types that we have learned about. A \_\_\_\_\_ can be used to store a sequence of characters, while a \_\_\_\_\_ can store a sequence of any type of data (but is immutable). A \_\_\_\_\_ can also store any type of data, and allows you to change elements within it. Finally, a \_\_\_\_\_ can associate a value to a key.

## 3. Fill in the blanks 2 (12 points):

The \_\_\_\_\_ function returns a sequence containing all of the pixels in a picture.

The \_\_\_\_\_ sort algorithm operates by first finding the smallest element of the array and then moving it to the front (position [0]) of the array.

The \_\_\_\_\_ sort algorithm runs in  $O(N^2)$  time.

The \_\_\_\_\_ sort algorithm runs in  $O(N \log N)$  time complexity.

The \_\_\_\_\_ search algorithm takes  $O(N)$  time to find a value, but has the advantage of working on unsorted data.

The \_\_\_\_\_ search algorithm requires data to be sorted.

## 4. Program Comprehension (4 points)

```
def n_lines(n):
    if n >= 0:
        print "Line!"
        n_lines(n-1)
```

How many times will the string "Line!" be printed when n\_lines is called with n=4?  
Number \_\_\_\_\_

**5. Read Code (8 Points):**

The following code is defined. Assume the `turnL45()` function turns the robot left (counter-clockwise from the top) exactly 45 degrees. Also assume that the `getObstacle("center")` function will return 500 or higher if the robot is facing any wall.

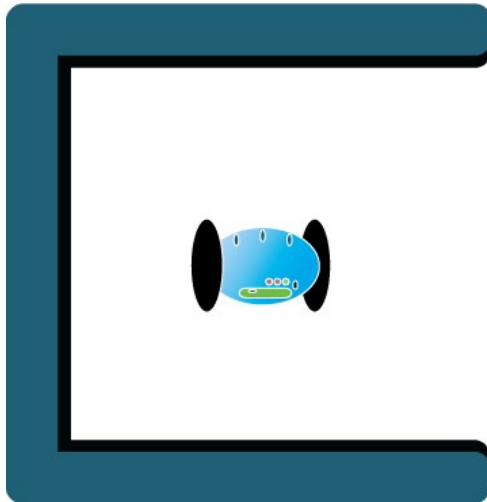
```
def turnL45():
    turnLeft(.5, 1)

def findPathOut():
    ir = getObstacle("center")
    out = False

    while out != True:
        if ir < 400:
            forward(1,5)
            out = True

        turnL45()
        ir = getObstacle("center")

findPathOut()
```



Please mark the wall(s) with the order that the robot faces them (and scans them with the Obstacle / IR sensors). The first wall it faces/scans should be marked with a 1, and so on, until the robot escapes.

Your Name: \_\_\_\_\_ Your TA's Name: \_\_\_\_\_

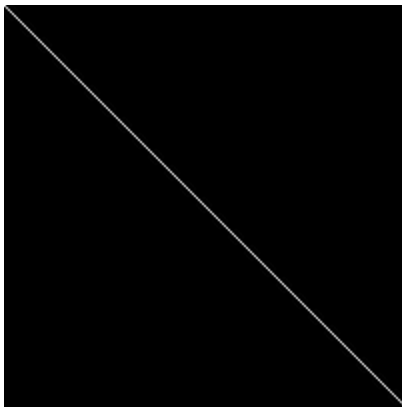
5/9

### 6. Reading Code: (8 points)

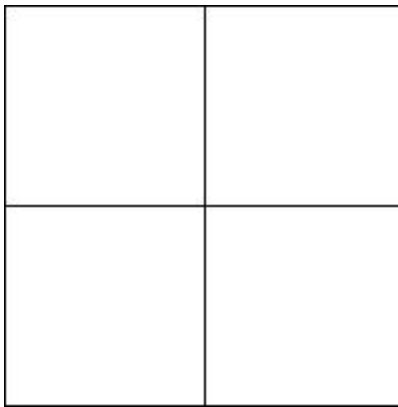
The following code is run in python:

```
picture = makePicture(201, 201)
for pixel in getPixels(picture):
    if (getX(pixel) % 100 == 0) | (getY(pixel) % 100 == 0):
        setRed(pixel, 0)
        setGreen(pixel, 0)
        setBlue(pixel, 0)
    else:
        setRed(pixel, 255)
        setGreen(pixel, 255)
        setBlue(pixel, 255)
show(picture)
```

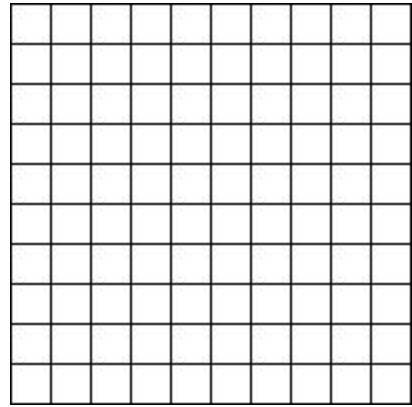
What image is shown? *Letter of image shown:* \_\_\_\_\_



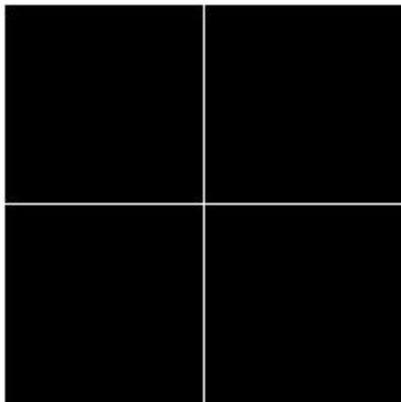
A



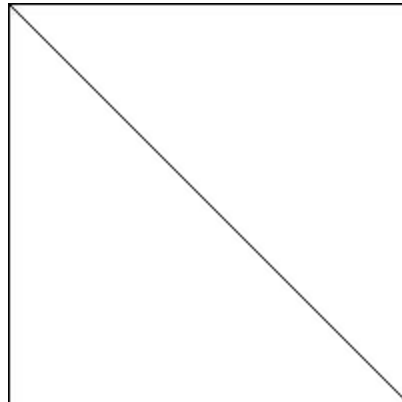
B



C



D



E

Your Name: \_\_\_\_\_ Your TA's Name: \_\_\_\_\_

6/9

## 7. Code Understanding: (9 points)

```
def function():  
    print "I often wonder which is mine"  
    return "Tolerance"  
    print "Or a rubber spine"
```

A. What appears in the console if we then run the following code from the console?

```
function()
```

B. What appears in the console if we then run the following code from the console?

```
myVar = function()
```

C. Assuming part B has been run, what do we see if we run the following code from the console?

```
print myVar
```

Your Name: \_\_\_\_\_ Your TA's Name: \_\_\_\_\_

7/9

### 8. Write Code! (12 points)

Write a function **printFile( fileName )** that will open the file name passed in the **fileName** parameter, read each line in the file, and print the contents of each line to the screen. You may assume that the file will exist and that you do not have to do error checking.

Your Name: \_\_\_\_\_ Your TA's Name: \_\_\_\_\_

8 / 9

## 9. Write Code (15 points)

Write a function named **maxValues(pic)** that takes in one parameter, a picture, and finds the highest blue, red, and green values in it. Print the maximum values to the console as specified by the example below. Nothing should be returned by the function. *Note: The maximum value for each color can occur in different pixels.*

Example usage and output:

```
>>> p = takePicture()
```

```
>>> maxValues(p)
```

```
Max Red: 152   Max Green: 203   Max Blue: 245
```

```
>>>
```



Your Name: \_\_\_\_\_ Your TA's Name: \_\_\_\_\_

9 / 9

**10. Write Code! (18 points):**

Write a function `color2gray( picture)` that accepts a picture as the parameter, converts it to grayscale, and also returns a reference to the newly grayed picture. To convert a picture to grayscale, set every pixel's R, G, and B color values equal to the green value. *Note that your function should be a modifier, in that it modifies the original picture, and does not work on a copy of it.*

Extra Credit (1 point each)

Name the primary fictional robot from the following books, TV shows or Movies:

Talmud _____	The Jetsons _____
Metropolis _____	Futurama _____
The Day the Earth Stood Still _____	Short Circuit _____
2001- A Space Odyssey _____	Knight Rider _____
Aliens & Aliens <sup>3</sup> _____	