HOMEWORK 1

Due Wednesday, 18 Jan, at 10am

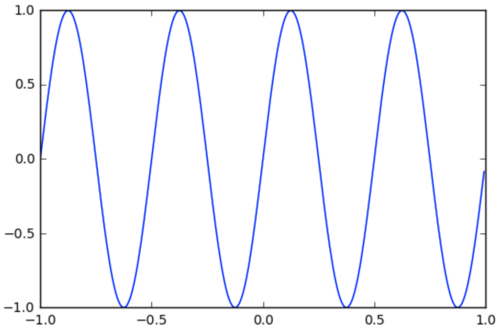
Please enter your answers into Homework01.ipynb and submit by the deadline via canvas.

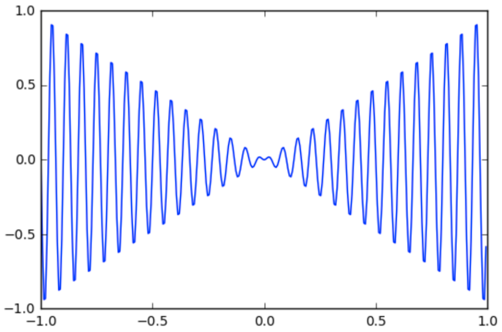
Problem 1. In Homework01.ipynb, you will see the following code

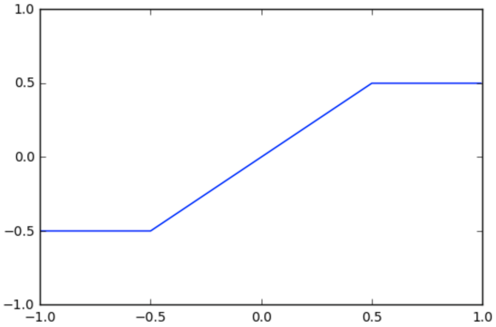
```
# libhw01 includes the drawing and graphing functions we are going to use
# note that libhw01.py needs to be in the same folder as the notebook file you
    are running
import libhw01 as hwlib
# can't do math without math can we?
import math

def f(x):
    return math.sin(math.pi*x)

hwlib.graphfunction(f)
```

Which produces the graph of the function f(x) = $\sin(\pi x)$

Write the functions that will produce the following graphs and use the `graphfunctions` function to graph them.



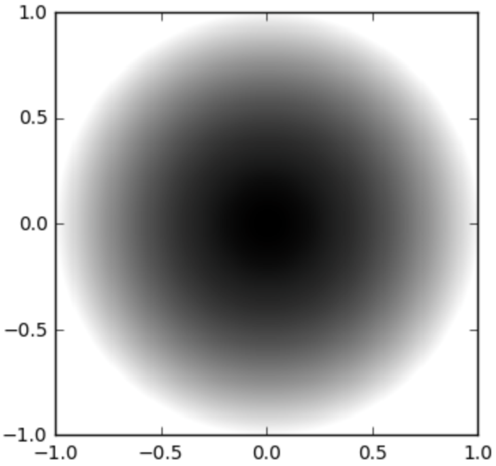




Problem 2. How do we graph a function $f(x, y)$ in two variables. One way to do it is to consider the function's value $f(x, y)$ at a point (x, y) as the color-intensity at that point. So, for example, if $f(0, 0) = 0$, then you put a black pixel at $(0, 0)$, i.e. the center. If $f(1, 1) = 1$, then you put a white pixel at (1,1), i.e. the top-right.

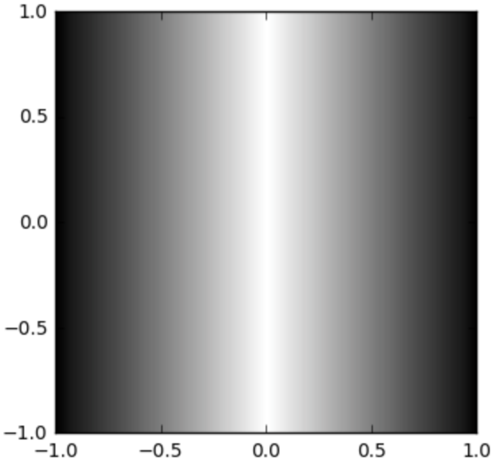
Consider the following code and the output:

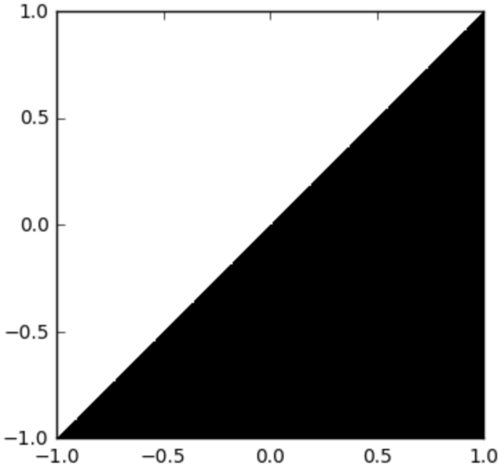
```
def fun1(x,y):
    return x**2 + y**2
hwlib.drawfunction(fun1)
```

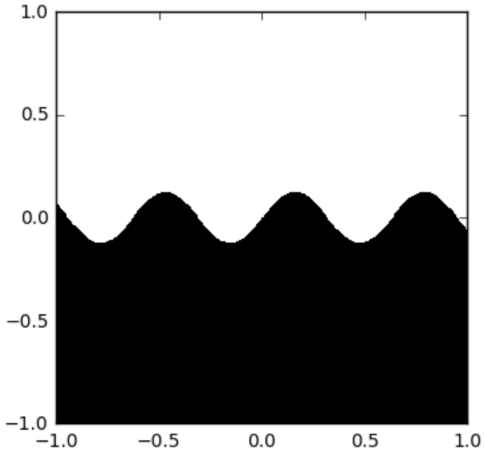


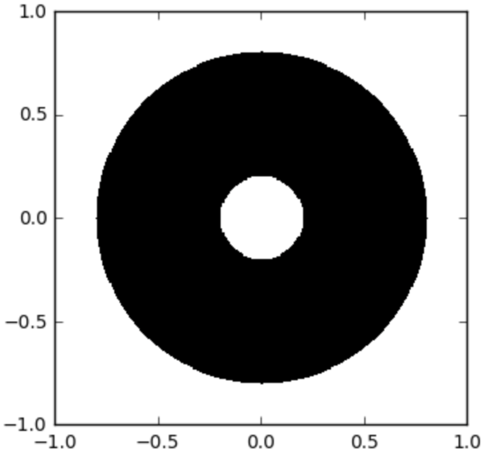
Note that values above 1 are still represented as white pixels, and values below 0 are represented as black pixels.

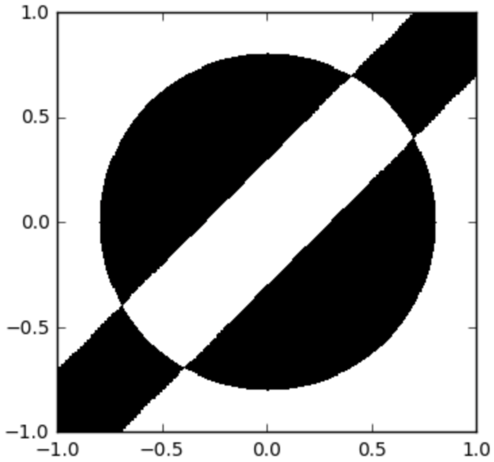
Write down functions that will produce the following pictures:

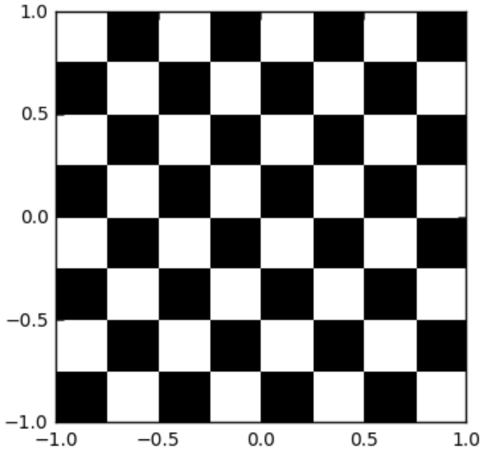












The chessboard is not easy. You may want to use the `math.floor` function. (optional: if you want a bigger challenge, make a function called `chessboard(n)`, which returns a function that, when graphed with drawfunction, will make an $n \times n$ chessboard)

Problem 3. Produce a picture you think is interesting. We will use these to challenge your colleages in the future.

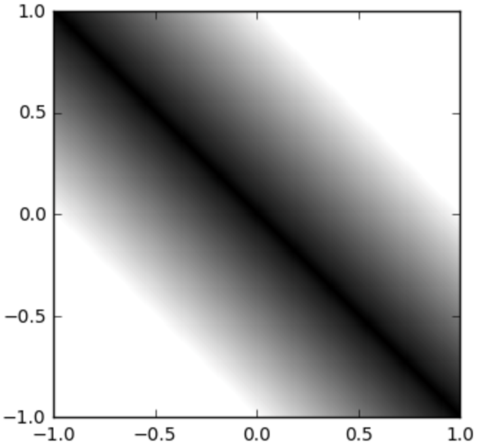
Problem 4. Functions in python can be used as inputs to a function and can ve returned just like everything else. For example, consider the function `rotateleft`

```
def rotateleft(f):
    def g(x,y):
        return f(y, -x)
    return g
```

It takes in a function and returns a different function which we can then use. For example:

```
def fun(x,y):
    return abs(x+y)
hwlib.drawfunction(fun)
```

outputs



What will the following code output? (first guess then try running the code).

```
hwlib.drawfunction(rotateleft(fun))
```

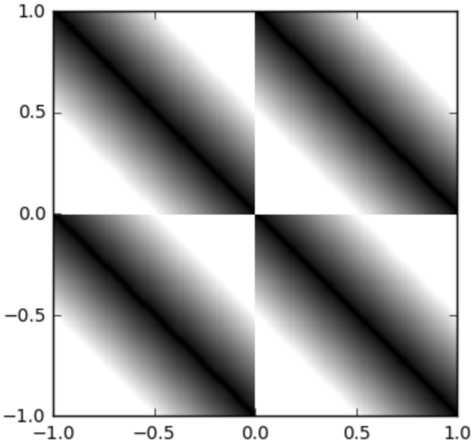
Modify the code for rotateleft so that it rotates to the left by $pi/3$ instead of $pi/2$. Then, write a function called repeat.

```
  def repeat(f):
    def g(x,y):
        ...
        return ...
    return g
```

such that when we call:

```
hwlib.drawfunction(repeat(fun))
```

We get


1.0
0.5
0.0
−0.5
−1.0
−1.0
−0.5
0.0
0.5
1.0