

# HONR 490 – Homework 1

Justin A. Gould  
gould29@purdue.edu

February 15, 2021

**DUE DATE:** 2021/08/29 23:59 EDT

## Homework Instructions

To receive credit for the assignment, do the following:

1. Create a `.py` file, and name it: `purduealias_honr490_homework_number.py`  
(e.g., `gould29_honr490_homework_1.py`)
2. Create a function for each problem, accepting the input and providing the desired output  
(both of which will be defined in the homework assignment).  
(e.g., `def problem_1()` for Problem #1)
3. Submit the `.py` file to Brightspace by the due date.

For grading, I will leverage unit tests, to ensure you aren't hard-coding your work. These unit tests are hidden. To test your code, I suggest using a Jupyter Notebook to ensure you're following directions. An example `.py` file is on our Brightspace and GitHub.

**Problem 1****Python Basics: Loops – 3 points**

For all non-negative integers  $i < n$ , print  $i^2$ .

For example, if  $n = 4$ :

The list of non-negative integers that are less than  $n = 4$  is:  $[0, 1, 2, 3]$

Add the square of each number to a list:  $[0, 1, 4, 9]$

**Input:** Any non-negative integer,  $n$ , with the following constraint(s):

- $0 \leq n < 20$

**Desired Output:** A list containing the square of each non-negative integer less than  $n$ .

**Problem 2****Python Basics: List Comprehension – 3 points**

You are given three integers,  $x, y, z$  representing the dimensions of a cuboid along with an integer  $n$ . Print a list of all possible coordinates given by  $(i, j, k)$  on a 3D grid where  $i + j + k \neq n$ . Here,  $0 \leq i \leq x; 0 \leq j \leq y; 0 \leq k \leq z$ . Please use list comprehensions rather than multiple loops, as a learning exercise. For example:

$x = 1$

$y = 1$

$z = 1$

$n = 2$

All permutations of  $[i, j, k]$  are:  $[[0, 0, 0], [0, 0, 1], [0, 1, 0], [1, 0, 0], [1, 1, 1]]$ .

**Input:** Four integers:  $x, y, z, n$

**Desired Output:** A list in lexicographic increasing order.

**The remaining questions will utilize the following datasets:**

- **meat:** Metrics on livestock, dairy, and poultry outlook and production
- **births:** Demographic statistics on live births by month.

These datasets can be loaded via the `pandasql` package (see the boilerplate `.py` file in the Homework 1 directory on GitHub!):

```
from pandasql import *  
meat = load_meat()  
births = load_births()
```

**NOTE:** For questions 3-6, please define the SQL query in your function, such as:

```
def problem_3(sql="YOUR QUERY"):
```

### Problem 3

**SQL Review: Basic Filtering and Retrieval (a) – 3 points** Please use SQL to answer the following question: How many dates have beef production > 750?

**Input:** SQL query text

**Desired Output:** Number (as integer) of dates where beef production > 750.

### Problem 4

**SQL Review: Basic Filtering and Retrieval (b) – 3 points** Please use SQL to answer the following question: What date has the highest beef production?

**Input:** SQL query text

**Desired Output:** Python `datetime` object of the date with the highest beef production.

### Problem 5

**SQL Review: Basic Filtering and Retrieval (c) – 3 points** Please use SQL to answer the following question: How many dates is turkey production NULL?

**Input:** SQL query text

**Desired Output:** Number (as integer) of dates where turkey production is NULL.

### Problem 6

**SQL Review: JOIN and Subquery – 5 points**

Please use SQL to answer the following question: What is the average beef production on the dates where there are more than 300,000 births?

**Input:** SQL query text

**Desired Output:** Float representing the average beef production, rounded 2 places.