Answer Key:

Mock Final Exam
CSci 127: Introduction to Computer Science
Hunter College, City University of New York

17 May 2022

1. (a) What will the following Python code print:

i.
```
a = "Jan&Feb&Mar&Apr&May&Jun"
print(a.count("&"))
```

Answer Key:
```
5
```

ii.
```
b = a.split("&")
print(b[0])
```

Answer Key:
```
Jan
```

iii.
```
mo = b[-1].upper()
print(mo)
```

Answer Key:
```
JUN
```

iv.
```
for c in mo:
    print(c.lower())
```

Answer Key:
```
j
u
n
```

(b) Consider the following shell commands:

```
$ ls -l
```

```
-rw-r--r--@ 1 ligorio  staff      5308 Mar 21 14:38 quizzes.html
-rw-r--r--  1 ligorio  staff       413 Apr 20 18:57 zoneDist.csv
-rw-r--r--@ 1 ligorio  staff      519 Apr 22 15:14 zoneMap.py
-rw-r--r--  1 ligorio  staff  16455174 Mar 20 19:02 zoning2.html
-rw-r--r--  1 ligorio  staff  17343896 Mar 20 18:58 zoningIDS.json
```

i. What is the output for:
```
$ ls *zz*
```

Answer Key:

```
quizzes.html
```

ii. What is the output for:
```
$ ls -l | grep "Apr"
```

Answer Key:

```
-rw-r--r--  1 ligorio  staff       413 Apr 20 18:57 zoneDist.csv
-rw-r--r--@ 1 ligorio  staff      519 Apr 22 15:14 zoneMap.py
```

iii. What is the output for:
```
$ ls -l | grep "Apr" | wc -l
```

Answer Key:

```
2
```

2. (a) For each row below containing a binary, decimal, and hexadecimal number, circle the **largest value** in the row (or "All Equal" if all three entries have the same value):

Answer Key:

	Binary:	Decimal:	Hexadecimal:	All Equal
a)	10	10	**10**	*All Equal*
b)	1100	12	C	**All Equal**
c)	10010	18	12	**All Equal**
d)	100000	**34**	19	*All Equal*
e)	1111110	250	**FE**	*All Equal*

(b) Fill in the code below to make an image in which a pixel is white if it has an entry of 0 in the array `elevations`. Otherwise, the pixel should be colored green.

```
# Takes elevation data of NYC and displays coastlines
import numpy as np
import matplotlib.pyplot as plt
elevations = np.loadtxt('elevationsNYC.txt')
#Base image size on shape (dimensions) of the elevations:
mapShape = elevations.shape + (3,)
floodMap = np.zeros(mapShape)

for row in range(mapShape[0]):
    for col in range(mapShape[1]):
```

Answer Key:

```
        if elevations[row,col] == 0:
            #Coastline:
            floodMap[row,col,:] = 1.0     #Set all channels to 100%
        else:
            #Everyone else
            floodMap[row,col,1] = 1.0   #Set the green channel to 100%

#Save the image:
plt.imsave('floodMap.png', floodMap)
```

3. (a) What is the value (True/False):

```
in1 = True
```

i.

```
in2 = False
out = (not in1) and (not in2)
```

Answer Key:

```
out = False
```

ii.

```
in1 = False
in2 = True
out = (not in1 or in2) and (not in2 or in1)
```

Answer Key:

```
out = False
```

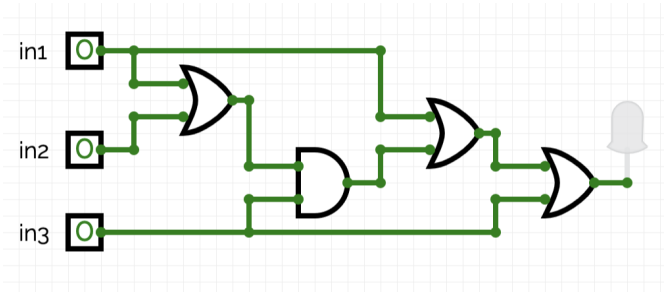
iii.

```
in1 = not False
in2 = not False or False
in3 = not in1 or not in2
out = not in2 and not in3
```

Answer Key:

```
out = False
```

iv.




```
in1 = False
in2 = True
in3 = False
```

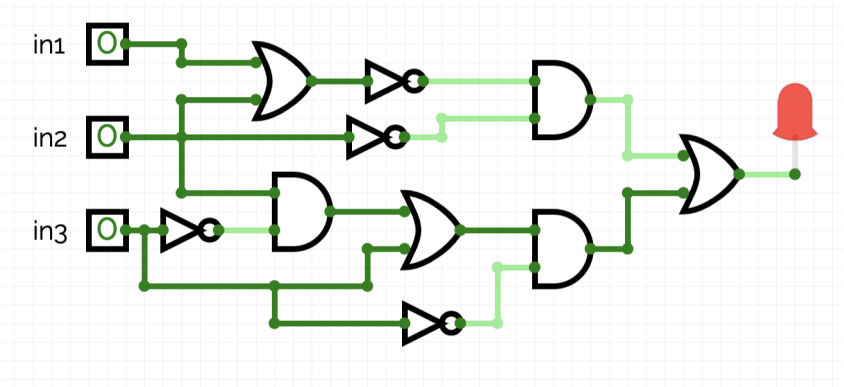
Answer Key:

```
out = False
```

(b) Design a circuit that implements the logical expression:

```
(not (in1 or in2) and (not in2)) or (((in2 and not in3) or in3) and not in3)
```

Answer Key:

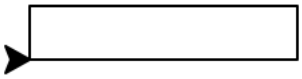


4. (a) Draw the output for the function calls:

```
import turtle

def mystery1(tess, x, y):
    for i in range(2):
        tess.forward(x)
        tess.left(90)
        tess.forward(y)
        tess.left(90)

def mystery2(tina, s):
    mystery1(tina, s, s)


taj = turtle.Turtle()
```

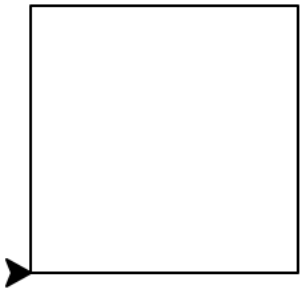
i. `mystery1(taj, 100, 20)`

Answer Key:



ii. `mystery2(taj, 100)`

Answer Key:



(b) Given the function definitions:

```
def enigma(n):
    for i in range(1,n+1):
        help(i)
        print()
```

```
def help(x):
    for j in range(x):
        print((x+j)*2,end=' ')
```

i. What is the output for `enigma(5)`?

Answer Key:

2
4 6
6 8 10
18 10 12 14
10 12 14 16 18

5. Design an algorithm that asks the user for the name of a text file containing a grid of numbers and loads it into a 2D array of integers (think like an image without the color channel), then outputs the index `(row, col)` of the SMALLEST number in the array.
Libraries:

Answer Key: numpy
Input:

Answer Key: The input file
Output:

Answer Key: The index of the smallest number
Design Pattern:

Answer Key: ☐ Search **X** Find Min ☐ Find Max ☐ Find All
Principal Mechanisms (select all that apply):

Answer Key: ☐ Search ☐ Single Loop **X** Nested Loop **X** Conditional (if/else) statement
XIndexing / Slicing ☐ `split()` **X** `input()`

Process (as a concise and precise LIST OF STEPS / pseudocode):
(Assume libraries have already been imported.)

Answer Key:

(a) Ask the user for input file name
(b) Load the data into a numpy array, call it `grid`
(c) Set variables min_row and min_col to 0

(d) Use a nested loop to consider every number in the `grid` looping for rows in outer loop and columns in inner loop

i. if the current number (the number at grid[current_row, current_column] < grid[min_row, min_col], set min_row to current_row and set min_col to current_column

(e) Return min_row and min_col

6. Write a **complete Python program** that asks the user for the name of a .png (image) file and displays the upper right quarter of the image.

   For example if the image is `hunterLogo.png` (left), the displayed image would be (right):

Answer Key:

```
#Name:  CSci 127 Teaching Staff
#Date:  Fall 2017
#This program loads an image and  creates and displays
#    a new image that is only the upper left corner.

#Import the packages for images and arrays:
import matplotlib.pyplot as plt
import numpy as np

inF = input('Enter file name: ')
img = plt.imread(inF)              #Read in image from inF

height = img.shape[0]              #Get height
width = img.shape[1]               #Get width
print(height,width)

img2 = img[:height/2, width/2:]    #Crop to lower left corner

plt.imshow(img2)                   #Load our new image into pyplot
plt.show()                         #Show the image (waits until closed to continue)
```

7. Fill in the following functions that are part of a program that maps GIS data from NYC OpenData CSV files:

   - `getData()`: asks the user for the name of the CSV and returns a DataFrame of the contents.

- `getLocale()`: asks the user for latitude and longitude of the user's current location and returns those floating points numbers, and
- `computeDist()`: computes the squared distance between two points (x1,y1) and (x2,y2):

$$(x1 - x2)^2 + (y1 - y2)^2$$

Answer Key:

```
import pandas as pd
def getData():
     """
     Asks the user for the name of the CSV and
     Returns a dataframe of the contents.
     """
     inF = input('Enter CSV file name: ')
     df = pd.read_csv(inF)
     return(df)

def getLocale():
     """
     Asks the user for latitude and longitude of the user's current location and
     Returns those floating points numbers.
     """
     lat = float(input('Enter current latitude: '))
     lon = float(input('Enter current longitude: '))
     return(lat, lon)

def computeDist(x1,y1,x2,y2):
     """
     Computes the squared distance between two points (x1,y1) and (x2,y2) and
     Returns (x1-x2)^2 + (y1-y2)^2
     """
     d = (x1 - x2)**2 + (y1 - y2)**2
     return(d)
```

8. (a) What is printed by the MIPS program below:

Answer Key:

```
!!!!!
```

(b) Modify the program to print out 99 copies of the character '!'. Shade in the box for each line that needs to be changed and rewrite the instruction below.

Answer Key:

```
#Loop through characters
ADDI $sp, $sp, -100      # Set up stack
ADDI $s3, $zero, 1       # Store 1 in a register
ADDI $t0, $zero, 33      # Set $t0 at 33 (!)
ADDI $s2, $zero, 99      # Use to test when you reach 100
SETUP: SB $t0, 0($sp)    # Next letter in $t0
ADDI $sp, $sp, 1         # Increment the stack
SUB $s2, $s2, $s3        # Decrease the counter by 1
BEQ $s2, $zero, DONE     # Jump to done if $s0 == 0
J SETUP                  # If not, jump back to SETUP for loop
DONE: ADDI $t0, $zero, 0 # Null (0) to terminate string
SB $t0, 0($sp)           # Add null to stack
ADDI $sp, $sp, -99      # Set up stack to print
ADDI $v0, $zero, 4       # 4 is for print string
ADDI $a0, $sp, 0         # Set $a0 to stack pointer for printing
syscall                  # Print to the log
```

9. Fill in the C++ programs below to produce the Output on the right.

```
#include <iostream>
using namespace std;
int main()
{

    for(int i = 0; i <=30; [          ]){
```

(a) **Answer Key:**

```
i += 10


       cout << i*2  << endl;
    }
    return 0;
}
```

```
#include <iostream>
using namespace std;
int main()
{
    int count = 5;
    int num = 2;
```

(b)

```
    while(count [ ] && num [ ]){
        cout << count << " " << num << endl;
        count -=1;
        if(count % 2 == 0)
            num -=1;
    }
    return 0;
}
```

Answer Key:

```
count > 0 \&\& num >= 0
or
count >= 1 \&\& num > -1
```

```
#include <iostream>
using namespace std;
int main(){

    for (int i = 5; [ ]; i--){
```

(c)

Answer Key:

```
i > -5
or
i >= -4
```

```
        cout << "Still counting!" << endl;
    }
    return 0;
}
```

10. (a) Translate the following program into a **complete C++ program**:

```
#Python Loops, V3:
for i in range(0,50,5):
    print(i)
```

Answer Key:

```
//C++ Loop, V3
#include <iostream>
```

```
using namespace std;
int main()
{
  int i;
    for (i = 0; i < 50; i=i+5) {
        cout << i << endl;
    }
  return 0;
}
```

(b) Write a **complete C++ program** to compute the ticket price to enter the Museum of Natural History. Your program must ask the user for their age and print "Child: $12.50" if the age entered is 12 or less, "Adult: $22.00" if the age entered is less than 65, and "Senior: $17.00" otherwise.

Answer Key:

```
//Prints ticket price for the Musemum of Natural History
#include <iostream>
using namespace std;
int main()
{
    cout << "Please enter your age: ";
    int age = 0;
    cin >> age;
    if (age <= 12)
        cout << "Child: $12.50\n";
    else if (age < 65)
        cout << "Adult: $22.00\n";
    else
        cout << "Senior: $17.00\n";
  return 0;
}
```