Final Exam, Version 1
CSci 127: Introduction to Computer Science
Hunter College, City University of New York

13 December 2018

Answer Key:

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

i.
```
s = "avram,henriette;dolciani,mary;rees,mina"
a = s[0:5]
print(a.upper())
```

Answer Key:

```
AVRAM
```

ii.
```
names = s.split(';')
print(names[-1])
```

Answer Key:

```
rees,mina
```

iii.
```
b,c = names[1],names[2]
print(c[-4:])
```

Answer Key:

```
mina
```

iv.
```
for n in names:
    w = n.split(',')
    print(w[1],w[0])
```

Answer Key:

```
henriette avram
mary dolciani
mina rees
```

(b) Consider the following shell commands:

```
$ ls -l
-rw-r--r--@  1 stjohn  staff     87772 May 19  2018 flower.png
-rw-r--r--@  1 stjohn  staff     13061 May 19  2018 lab1.html
-rw-r--r--@  1 stjohn  staff     13481 Jun 19 14:25 lab6.html
-rw-r--r--@  1 stjohn  staff      8698 Jul  9 08:52 lab7.html
-rw-r--r--@  1 stjohn  staff     10745 Jul 28  2017 star.png
```

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

Answer Key:

```
lab1.html  lab6.html  lab7.html
```

ii. What is the output for:
`$ ls *.html | wc -l`

Answer Key:

```
3
```

iii. What is the output for:
`$ ls -l | grep "Jul" | 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)	1	1	1	**All Equal**
b)	11	11	**11**	*All Equal*
c)	**11111**	29	1C	*All Equal*
d)	100000	**34**	20	*All Equal*
e)	11111111	255	FF	**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 red.

```
# 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,0] = 1.0   #Set the red channel to 100%

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

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

i.
```
in1 = True
in2 = False
out = in1 and in2
```

Answer Key:
`out = False`

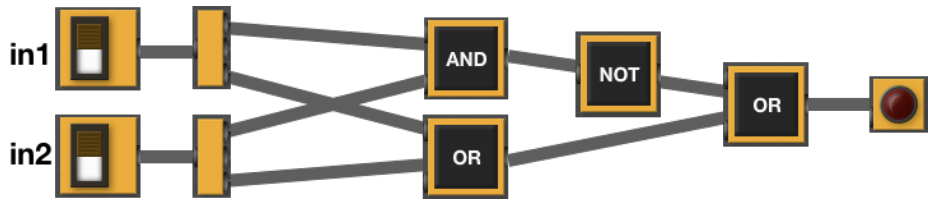
ii.
```
in1 = False
in2 = False
out = not in1 and (in2 or not in1)
```

Answer Key:
`out = True`

iii.
```
in1 = True
in2 = False and not in1
in3 = in1 and in2
out = in1 and not in3
```

Answer Key:
`out = True`

iv.



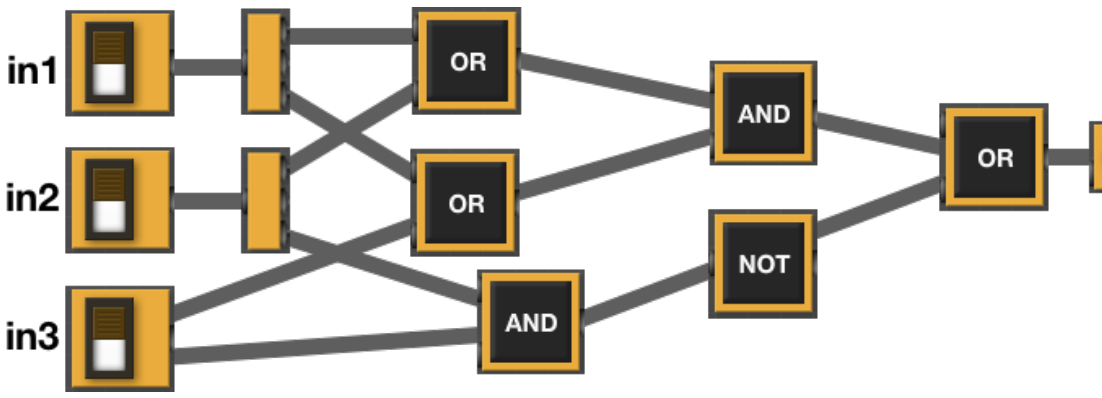

```
in1 = True
in2 = False
```

Answer Key:
`out = True`

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

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

Answer Key:



4. (a) For the following code:

```
def v1(maria, lily):
    if  maria < lily:
        return maria
    else:
        return -1
```

```
def startV1(such):
    alex = 5
    jaime = 20
    dandan = v1(alex+such,jaime)
    return dandan
```

i. What are the formal parameters for `v1()`:

Answer Key:
`maria, lily`

ii. What are the formal parameters for `startV1()`:

Answer Key:
`such`

iii. What does `startV1(30)` return:

Answer Key:
`-1`

(b) Given the function definition:

```
def sorted(ls):
    for i in range(3):
        print(ls)
        for j in range(2):
            if ls[j] > ls[j+1]:
                 ls[j],ls[j+1] = ls[j+1],ls[j]
```

i. What is the output for `sorted([9,8,1,3])`?

Answer Key:

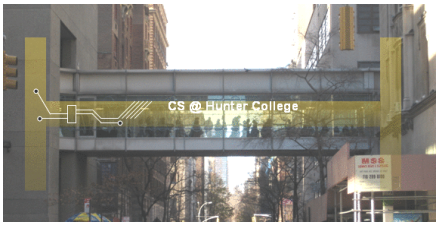
ls[0]	ls[1]	ls[2]	ls[3]
9	8	1	3
8	1	9	3
1	8	9	3

ii. What is the output for `sorted(["Matt","Ilana","Carol","Ally"])`?

Answer Key:

ls[0]	ls[1]	ls[2]	ls[3]
"Matt"	"Ilana"	"Carol"	"Ally"
"Ilana"	"Carol"	"Ally"	"Matt"
"Carol"	"Ally"	"Ilana"	"Matt"

5. Design an algorithm that asks the user for an input image name, an output image name, and then crops the image to dimensions given by the user and saves to the new file. Specify the inputs and outputs for your algorithm and give the design in pseudocode. In your pseudocode, specify any libraries that you would need for your design.



Input:

Answer Key: The name of the input image and output image files and the upper, lower, left, and right dimensions for cropping.

Output:

Answer Key: The cropped image (saved to the file specified by the user).

Process:

Answer Key:

(a) Import numpy and pyplot.
(b) Ask user for file names and dimensions for cropping.
(c) Save input file to an array.
(d) Copy the cropped portion to a new array.
(e) Save the new array to the output file.

6. Fill in the Python program that will read:

- prompt the user for the name of a CSV file,
- prompt the user for the name of a column in that CSV file, and
- print out the minimum value of the column.
- displays a scatter plot of `"Year"` column versus the column entered.

```
#P6,V1: prints max of a column in a CSV file & makes a scatter plot

#Import the libraries for data frames and displaying images:
```

```
#Prompt user for file name:
```

```
#Prompt user for column name:
```

```
df = pd.read_csv(fileName)
#Compute maximum value of the column:
```

```
print("Maximum is ", M)

#Display a scatter plot of "Year" vs. column entered by user:
```

Answer Key:

```
#P6,V1: prints max of a column in a CSV file & makes a scatter plot

#Import the libraries for data frames and displaying images:
import pandas as pd
import matplotlib.pyplot as plt
```

```
#Prompt user for file name:
fileName = input('Enter file name: ')

#Prompt user for column name:
col = input('Enter column name: ')

df = pd.read_csv(fileName)

#Compute maximum value of the column:
M = df[col].max()
print("Maximum is ", M)

#Display a scatter plot of "Year" vs. column entered by user:
df.plot.scatter(x = "Year", y = col)
plt.show()
```

7. Complete the following program, by writing the functions:

   - `getInput()`: returns the number of turtles the user entered
   - `setUpTurtles()`: creates and returns a list of turtles, and
   - `moveForward()`: moves each turtle in the list forward.

   Answer Key:

```
#Intro Programming Lab:  A program with herd of turtles
import turtle
def getInput():
    """
    Prompts & returns the number of turtles the user entered.
    """
    n = eval(input("Please enter the number of turtles: "))
    return n
def setUpTurtles(n):
    """
    Creates a list of n turtles and returns the list.
    """
    tList = []
    #Create turtles:
    for i in range(n):
        t = turtle.Turtle()
        t.shape("turtle")       #Make the turtle appear turtle-shaped
        tList.append(t)
    return tList
def moveForward(tList):
    """
    Move each turtle in the list forward 30 steps.
    """
```

```
    for t in tList:
        t.forward(30)
def stamp(tList):
    for t in tList:
        t.stamp()
def main():
    numTurtles = getInput()     #Ask for number of turtles
    turtleList = setUpTurtles(numTurtles) #Make a list of turtles
    for i in range(10):
        moveForward(turtleList) #Move each turtle in the list forward
        stamp(turtleList)       #Stamp where the turtle stopped
if __name__ == "__main__":
    main()
```

8. (a) What is the output for a run of this MIPS program:

```
# Store message on the stack
ADDI $sp, $sp, -8
ADDI $t0, $zero, 72 # H
SB $t0, 0($sp)
ADDI $t0, $zero, 117 # u
SB $t0, 1($sp)
ADDI $t0, $zero, 110 # n
SB $t0, 2($sp)
ADDI $t0, $zero, 116 # t
SB $t0, 3($sp)
ADDI $t0, $zero, 101 # e
SB $t0, 4($sp)
ADDI $t0, $zero, 114 # r
SB $t0, 5($sp)
ADDI $t0, $zero, 33 # !
SB $t0, 6($sp)
ADDI $t0, $zero, 0 # (null)
SB $t0, 7($sp)

ADDI $v0, $zero, 4 # 4 is for print string
ADDI $a0, $sp, 0
syscall  # print to the log
```

Answer Key:

```
Hunter!
```

(b) Write a MIPS program that prints: `Hello!!`

Answer Key:

```
# Store message on the stack
```

```
ADDI $sp, $sp, -8
ADDI $t0, $zero, 72 # H
SB $t0, 0($sp)
ADDI $t0, $zero, 101 # e
SB $t0, 1($sp)
ADDI $t0, $zero, 108 # l
SB $t0, 2($sp)
ADDI $t0, $zero, 108 # l
SB $t0, 3($sp)
ADDI $t0, $zero, 111 # o
SB $t0, 4($sp)
ADDI $t0, $zero, 33 # !
SB $t0, 5($sp)
ADDI $t0, $zero, 33 # !
SB $t0, 6($sp)
ADDI $t0, $zero, 0 # (null)
SB $t0, 7($sp)

ADDI $v0, $zero, 4 # 4 is for print string
ADDI $a0, $sp, 0
syscall  # print to the log
```

9. What is the output of the following C++ programs?

(a)

```
//Lyrics by Lopez & Lopez
#include <iostream>
using namespace std;
int main()
{
  cout << "It's funny how some ";
  cout << "distance\nMakes";
  cout << "everything seem small";
  cout << endl;
  return(0);
}
```

Answer Key:

```
It's funny how some distance
Makes everything seem small
```

(b)
```
//More Elsa
#include <iostream>
using namespace std;
int main()
{
  int count = 2;
  while (count > 0) {
    cout <<"Let it go, ";
    count--;
  }
  cout << "\nCan't hold it ";
  cout << "back anymore\n";
  return(0);
}
```

Answer Key:

```
Let it go, let it go.
Can't hold it back anymore
```

(c)
```
//Stars and srtipes
#include <iostream>
using namespace std;
int main()
{
  int i, j;
  for (i = 0; i < 5; i++)
  {
    for (j = 0; j < 5; j++)
       if ( i % 2 == 0)
         cout << "*";
       else
         cout << "-";
    cout << endl;
  }
  return(0);
}
```

Answer Key:

```
*****
-----
*****
-----
*****
```

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

```
#Python Loop, V1:
```

```
for i in range(5,101,5):
    print(i)
```

Answer Key:

```
//C++ Loop, V1
#include <iostream>
using namespace std;
int main()
{
  int i;
  for (i = 5; i < 101; i = i + 5) {
    cout << i << endl;
  }
  return 0;
}
```

(b) Write a **complete C++ program** that prints the change in population of predator and prey following the Lotka-Volterra model:

$$\begin{aligned} r &= 1.5r - .2rf \\ f &= 0.95f + .1rf \end{aligned}$$

where r is the number of prey (such as rabbits) each year and f is the number of predators (such as foxes) each year. The rabbit population increases by 50% each year, but $\frac{rf}{5}$ are eaten by foxes. The fox population decreases by 5% due to old age but increases in proportion to the food supply, $\frac{rf}{10}$. Assume that the starting population of prey (rabbits) is 100 and starting population of predators (foxes) is 20. Your program should print for the first 10 years: the year, the number of prey and the number of predators.

Answer Key:

```
//Predator/Prey model
#include <iostream>
using namespace std;
int main()
{
  float r = 100, f = 20;
  int year;
  cout << "Year\tPrey\tPredators\n";
  for (year = 0; year < 10; year++) {
    cout << "\t" << year << "\t" << r << "\t" << f << "\n";
    r = 1.5*r - .2*r*f;
    f = 0.95*f + .1*r*f;
  }
  return 0;
}
```