Final Exam, Version 2
CSci 127: Introduction to Computer Science
Hunter College, City University of New York

19 December 2018

Answer Key:

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

i.
```
s = "elion,gertrude;cohn,mildred;petters,arlie"
a = s[0:5]
print(a.upper())
```

Answer Key:
```
ELION
```

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

Answer Key:
```
petters,arlie
```

iii.
```
b,c = names[1],names[2]
print(c[-5:])
```

Answer Key:
```
arlie
```

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
nyc.csv  p40.py  p41.py  p55.cpp  trees.csv
```

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

Answer Key:

```
nyc.csv  trees.csv
```

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

Answer Key:

```
2
```

iii. What is the output for:

```
$ mkdir data
$ echo "Created folder: data"
```

Answer Key:

```
Created folder: data
```

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)	**11**	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,0,:] = 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):

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

Answer Key:

```
out = True
```

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

Answer Key:

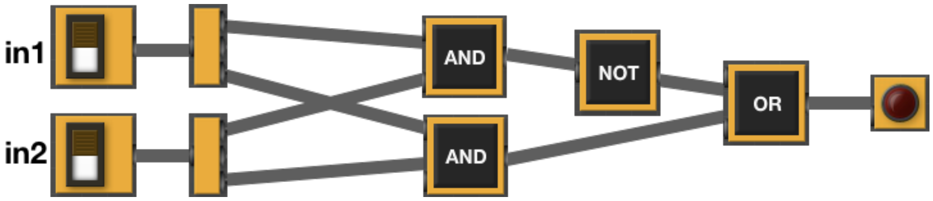
```
out = True
```

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

Answer Key:

```
out = True
```

iv.



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

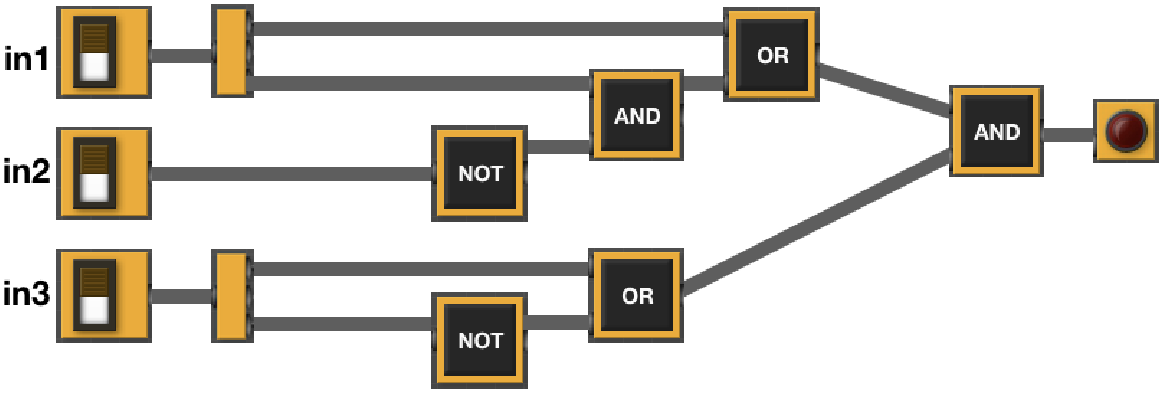
Answer Key:

```
out = True
```

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

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

Answer Key:



4. (a) For the following code:

```
def v2(ally, ilana):
    if  ally < ilana:
        return 0
    else:
        return ally
```

```
def startV2(antonio):
    david = 10
    melissa = 20
    saif = v1(antonio,melissa - david)
    return saif
```

i. What are the formal parameters for `v2()`:

Answer Key:
`ally, ilana`

ii. What are the formal parameters for `startV2()`:

Answer Key:
`antonio`

iii. What does `startV2(15)` return:

Answer Key:
`15`

(b) Given the function definition:

```
def sorted(ls):
    for i in range(4):
        print(ls)
        for j in range(3):
            if ls[j] > ls[j+1]:
                ls[j],ls[j+1] = ls[j+1],ls[j]
```

i. What is the output for `sorted([12,10,2,5])`?

Answer Key:

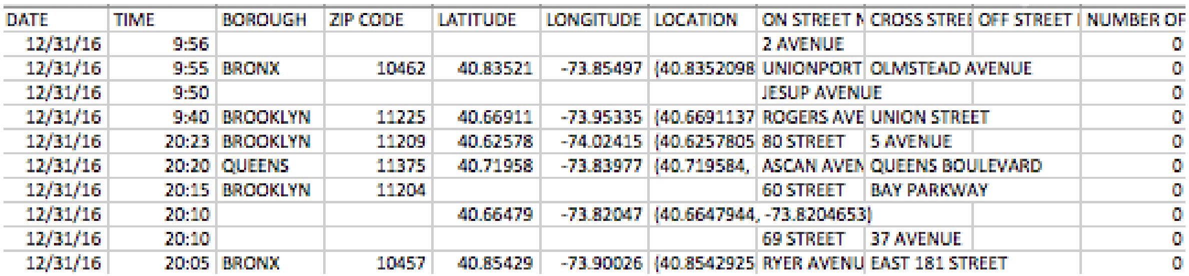
ls[0]	ls[1]	ls[2]	ls[3]
10	2	5	12
2	5	10	12
2	5	10	12
2	5	10	12

ii. What is the output for `sorted(["Yasmeen","Tommi","Owen","John"])`?

Answer Key:

ls[0]	ls[1]	ls[2]	ls[3]
"Yasmeen"	"Tommi"	"Owen"	"John"
"Tommi"	"Owen"	"John"	"Yasmeen"
"Owen"	"John"	"Tommi"	"Yasmeen"
"John"	"Owen"	"Tommi"	"Yasmeen"

5. Design an algorithm that prints out all the collisions in your zip code from the NYC Collisions OpenData. 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.



DATE	TIME	BOROUGH	ZIP CODE	LATITUDE	LONGITUDE	LOCATION	ON STREET N	CROSS STREE	OFF STREET I	NUMBER OF
12/31/16	9:56						2 AVENUE			0
12/31/16	9:55	BRONX	10462	40.83521	-73.85497	(40.8352098	UNIONPORT	OLMSTEAD AVENUE		0
12/31/16	9:50						JESUP AVENUE			0
12/31/16	9:40	BROOKLYN	11225	40.66911	-73.95335	(40.6691137	ROGERS AVE	UNION STREET		0
12/31/16	20:23	BROOKLYN	11209	40.62578	-74.02415	(40.6257805	80 STREET	5 AVENUE		0
12/31/16	20:20	QUEENS	11375	40.71958	-73.83977	(40.719584,	ASCAN AVEN	QUEENS BOULEVARD		0
12/31/16	20:15	BROOKLYN	11204				60 STREET	BAY PARKWAY		0
12/31/16	20:10			40.66479	-73.82047	(40.6647944, -73.8204653)				0
12/31/16	20:10						69 STREET	37 AVENUE		0
12/31/16	20:05	BRONX	10457	40.85429	-73.90026	(40.8542925	RYER AVENU	EAST 181 STREET		0

Input:

Answer Key: The name of the CSV file and the zip code

Output:

Answer Key: All trees (either rows or `Job ID`) of jobs containing “intern”.

Process:

Answer Key:

(a) Import pandas.

(b) Ask user for file name and zip code.

(c) Open the file as a dataFrame.

(d) Select all the rows (or `LOCATION`'s) where `ZIP CODE` matches the one entered.

(e) Print out selected rows (or `LOCATION`'s).

6. Fill in the Python program that will:

   - prompt the user for the name of a CSV file,
   - prompt the user for the name of a column in that CSV file,
   - print out the minimum value of the column, and
   - displays a bar plot of the column entered (with `"Year"` as the x-axis).

```
#P6,V2: prints min of a column in a CSV file & makes a 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 minimum value of the column:
```

```
print("Minimum is ", m)

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

Answer Key:

```
#P6,V2: prints min of a column in a CSV file & makes 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].min()
print("Minimum is ", m)

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

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

   - `getInput()`: returns the number of turtles the user entered,
   - `setUp()`: sets up a graphics window and turtle, and
   - `drawLines()`: repeat n times: 100 steps, turn left 92 degrees.

   Answer Key:

```
import turtle

def getInput():
    n = int(input("Enter number: "))
    return(n)

def setUp():
    trey = turtle.Turtle()
    win = turtle.Screen()
    return(win,trey)

def drawLines(t,n):
    for i in range(n):
        t.forward(100)
        t.left(92)

def main():
    n = getInput()  #get number of lines to be drawn
    w,t = setUp()   #sets up a graphics window and turtle
    drawLines(t,n)  #repeat n times: 100 steps, turn left 92 degrees

if __name__ == '__main__':
    main()
```

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

```
#Loop through first 5 letters:
ADDI $sp, $sp, -6   # Set up stack
ADDI $t0, $zero, 97 # Start $t0 at 97 (a)
ADDI $s2, $zero, 102 # Use to test when you reach 102 (f)
SETUP: SB $t0, 0($sp) # Next letter in $t0
ADDI $sp, $sp, 1    # Increment the stack
ADDI $t0, $t0, 1    # Increment the letter
BEQ $t0, $s2, DONE  # Jump to done if $t0 == 102
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, -6   # 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
```

Answer Key:

```
ABCDE
```

(b) Indicate What modifications are needed to the MIPS program (repeated below) so that it prints out the first 10 lower case letters: `abcdefghij` ?

Answer Key: Need to change:

- the first line to have space for 11 characters (ab...j and the null to terminate).
- the corresponding line to allow 11 characters to print (i.e. `ADDI $sp, $sp, -11`).
- the value of $s2 to be 107 (k).

The resulting program:

```
#Loop through first 10 letters:
ADDI $sp, $sp, -11   # Set up stack
ADDI $t0, $zero, 97 # Start $t0 at 97 (a)
ADDI $s2, $zero, 107 # Use to test when you reach 107 (k)
SETUP: SB $t0, 0($sp) # Next letter in $t0
ADDI $sp, $sp, 1    # Increment the stack
ADDI $t0, $t0, 1    # Increment the letter
BEQ $t0, $s2, DONE  # Jump to done if $t0 == 107
```

```
J SETUP             # If not, jump back to SETUP for loop
DONE: ADDI $t0, $zero, 0 # Null to end the string
SB $t0, 0($sp)      # Add null to stack
ADDI $sp, $sp, -11   # 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. What is the output of the following C++ programs?

(a)

```
//Lyrics by Lopez & Lopez
#include <iostream>
using namespace std;
int main()
{
  cout << "It's time to see what ";
  cout << "I can do\nTo test the";
  cout << "limits and break through";
  cout << endl;
  return(0);
}
```

Answer Key:

```
It's time to see what I can do
To test the limits and break through
```

(b)

```
//More Elsa
#include <iostream>
using namespace std;
int main()
{
  int count = 2;
  while (count > 0) {
    cout <<"Let it go, ";
    count--;
  }
  cout << "\nI am one with ";
  cout << "the wind and sky\n";
  return(0);
}
```

Answer Key:

```
Let it go, let it go,
I am one with the wind and sky
```

(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 ( j % 2 == 0)
         cout << "*";
       else
         cout << "-";
    cout << endl;
  }
  return(0);
}
```

Answer Key:

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

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

```
#Python Loops, V2:
for i in range(100,0,-5):
    print(i)
```

Answer Key:

```
//C++ Loop, V1
#include <iostream>
using namespace std;
int main()
{
  int i;
  for (i = 100; i > 0; i=i-5) {
    cout << i << endl;
  }
  return 0;
}
```

(b) Write a **complete C++ program** that asks the user for a whole number between -31 and 31 and prints out the number in "two's complement" notation, using the following algorithm:

i. Ask the user for a number, n.
ii. If the number is negative, print a 1 and let x = 32 + n.
iii. If the number is not negative, print a 0 and let x = n.
iv. Let b = 16.
v. While $b > 0.5$:
   - If $x >= b$ then print 1, otherwise print 0
   - Let x be the remainder of dividing x by b.
   - Let b be b/2.

Answer Key:

```
//twos complement
#include <iostream>
using namespace std;
int main()
{
    int n, x, b;
    cout << "Please enter a wole number between -31 and 31: ";
    cin >> n;

    if (n < 0) {
        cout << 1;
        x = 32 + n;
    } else {
        cout << 0;
        x = n;
    }

    b = 16;

    while (b > 0.5) {
        if (x >= b)
            cout << 1;
        else
            cout << 0;
        x = x % b;
        b = b / 2;
    }
    return 0;
}
```