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

19 December 2018

Exam Rules

- Show all your work. Your grade will be based on the work shown.
- The exam is closed book and closed notes with the exception of an 8 1/2" x 11" piece of paper filled with notes, programs, etc.
- When taking the exam, you may have with you pens and pencils, and your note sheet.
- You may not use a computer, calculator, tablet, smart watch, or other electronic device.
- Do not open this exam until instructed to do so.

Hunter College regards acts of academic dishonesty (e.g., plagiarism, cheating on examinations, obtaining unfair advantage, and falsification of records and official documents) as serious offenses against the values of intellectual honesty. The College is committed to enforcing the CUNY Policy on Academic Integrity and will pursue cases of academic dishonesty according to the Hunter College Academic Integrity Procedures.

I understand that all cases of academic dishonesty will be reported to the Dean of Students and will result in sanctions.
Name:
EmpID:
Email:
Signature:

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

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

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

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

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

(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
```
Output:

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

iii. What is the output for:
```
$ mkdir data
$ echo "Created folder: data"
```
Output:

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):

	Binary:	Decimal:	Hexadecimal:	All Equal
a)	11	2	2	*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]):









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

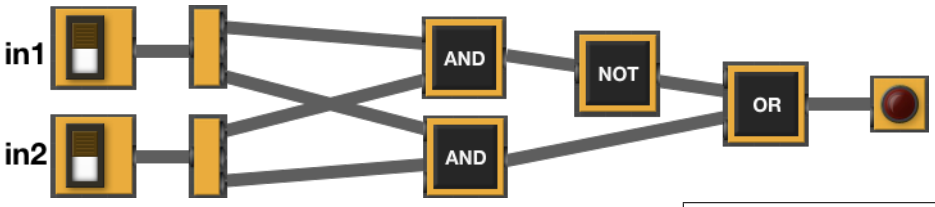
3. (a) What is the value (True/False):

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

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

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

iv.




```
in1 = True
in2 = True
```
out = ____

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

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

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()`:

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

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

(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])`?

ls[0]	ls[1]	ls[2]	ls[3]

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

ls[0]	ls[1]	ls[2]	ls[3]

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	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:

Output:

Process:

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:
```

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.

```
import turtle
def getInput():
     """
     Prompts for a whole number.
     Returns the number entered.
     """
```

```
def setUp():
     """
     Creates a graphics window and turtle.  Returns both.
     """
```

```
def drawLines(t,n):
     """
     Takes a turtle and n as input.
     Repeats n times: 100 steps, turn left 92 degrees.
     """
```

```
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
```

Output:

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

```
#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
```

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);
}
```

Output:

(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);
}
```

Output:

(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);
}
```

Output:

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

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

(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.