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

6 December 2017

Exam Rules

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

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

```
places = "Connecticut*New York*New Jersey*Pennsylvania"
num = places.count("*")
states = places.split("*")
print("There are", num+1, "states")
print(states[0], states[-1])
mess = "cahbecedseef gchaikje"
eat = ""
for i in range(len(mess)):
    if i % 2 == 0:
        print(mess[i])
        eat = eat + mess[i]
print("I love", states[1], eat)
```

Answer Key:

```
There are 4 states
Connecticut Pennsylvania
c
h
e
e
s
e

c
a
k
e
I love New York cheese cake
```

(b) Consider the following shell commands:

```
$ ls
p1.py  p2.py  p3.py
```

i. What is the output for:

```
$ cp p1.py p4.py
$ mv p2.py p5.py
$ ls p*
```

Answer Key:

```
p1.py  p3.py  p4.py  p5.py
```

ii. What is the output for:

```
$ mkdir programs
$ mv p*.py programs
$ ls
```

Answer Key:

```
programs/
```

2. (a) Fill in the missing values in the table:

Decimal	**Binary**	**Hexadecimal**
2	**Answer Key:** 10	2
Answer Key: 5	101	5
Answer Key: 10	1010	A
32	100000	**Answer Key:** 20
255	11111111	**Answer Key:** FF

(b) Write the Python code for the following algorithm:

```
function  decodeMessage(numbers, k)
    create an empty message
    for each number in the list of numbers
        code = 97 + ((number + k) modulo 26)
        convert the code to the corresponding Unicode character
        concatenate the character to the beginning of the message
    return the message
```

Answer Key:

```
def decodeMessage(numbers, k):
    mess = ""
    for number in numbers:
        code = 97 + ((number + k)%26)
        ch = chr(code)
        mess = mess + ch
    return(mess)
```

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

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

Answer Key:

```
out = False
```

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

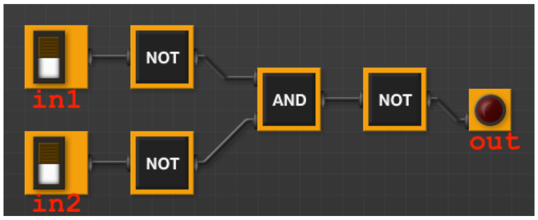
Answer Key:

```
out = True
```

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

Answer Key:

```
out = True
```

iv. 

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

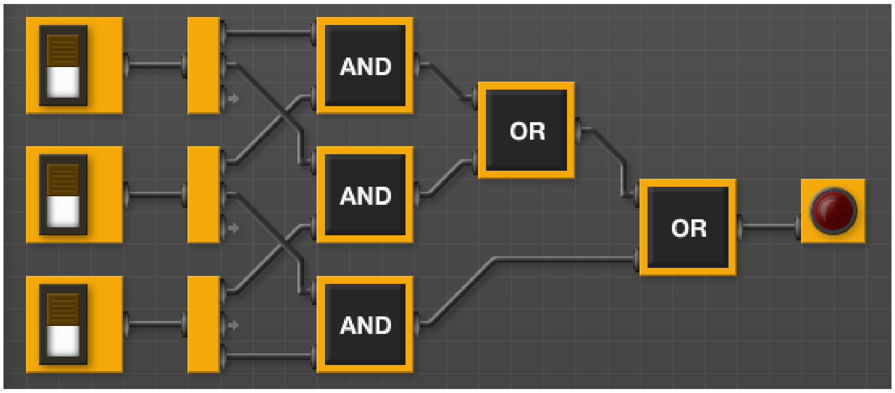
Answer Key:

```
out = True
```

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

```
(in1 and in2) or (in1 and in3) or (in2 and in3)
```

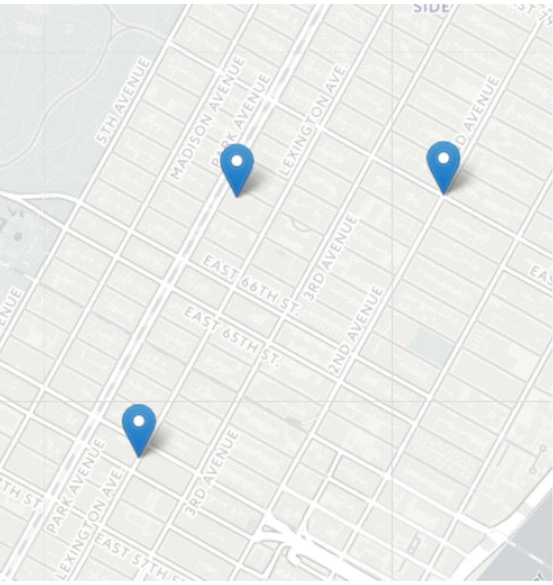
Answer Key:



4. (a) Draw the output of the program:

```
import folium as f
mapNYC = f.Map(location=[40.768731, -73.964915],\
               tiles="cartodbpositron",\
               zoom_start=15)
#68 & Lex:
f.Marker(location = [40.768731, -73.964915],\
         popup = "Hunter College").add_to(mapNYC)
#72 & 2nd:
f.Marker(location = [40.768802, -73.958361],
         popup = "Q Train").add_to(mapNYC)
#59 & Lex:
f.Marker(location = [40.762526, -73.967967],\
         popup = "4 Train").add_to(mapNYC)
mapNYC.save(outfile='nycMap.html')
```

Answer Key:



(b) What is returned when the function is invoked on the inputs below:

```
def enigma4(x,y,z):
    if x < y:
        return(z)
    elif x == y:
        return(z+z)
    else:
        s = cont4(z)
        return(s)
```

```
def cont4(st):
    r = ""
    for i in range(len(st)-1,-1,-1):
        r = r + st[i]
    return(r)
```

i. `enigma4(3,4,"blondie")`

Answer Key: `"blondie"`

ii. `enigma4(50,50,"Yum")`

Answer Key: `"YumYum"`

iii. `enigma4(10,0,"einworb")`

Answer Key: `"brownie"`

5. Write a **complete Python program** that will read:

   - prompt the user for the name of a CSV file,
   - prompt the user for the name of a column containing the x values,
   - prompt the user for the name of a column containing the y values, and
   - make a graph (using `matplotlib.pyplot`) of the x versus y values.

   Answer Key:

```
import matplotlib.pyplot as plt
import pandas as pd

inF = input(’Enter file name: ’)
data = pd.read_csv(inF)
xCol = input(’Enter column name for x values: ’)
yCol = input(’Enter column name for y values: ’)
data.plot(x = xCol, y = yCol)

plt.show()
```

6. Using `matplotlib.pyplot` and `numpy`, write a **complete Python program** that asks the user for the name of an image file, reads in the file, and displays only the “green” channel of the image to the screen.

   Answer Key:

```
#Name:  CSci 127 Teaching Staff
#Date:  Fall 2017
#This program loads an image, displays it, and then creates, displays,
#    and saves a new image that has only the red channel displayed.

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

inF = input('Enter image name: ')
img = plt.imread(inF)   #Read in image, inF
img2 = img.copy()        #make a copy of our image
img2[:,:,0] = 0          #Set the red channel to 0
img2[:,:,2] = 0          #Set the blue channel to 0

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,nrows=1000)
     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 the output for a run of this MIPS program:

```
# Store ’Hello!’ at the top of the stack
ADDI $sp, $sp, -7
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, 0 # (null)
SB $t0, 6($sp)
ADDI $v0, $zero, 4 # 4 is for print string
ADDI $a0, $sp, 0
syscall  # print to the log
```

Answer Key:

```
Hello!
```

(b) Write a MIPS program that prints: `Hi mom`

Answer Key:

```
# Store ’Hi mom’ at the top of the stack
ADDI $sp, $sp, -7
ADDI $t0, $zero, 72 # H
SB $t0, 0($sp)
ADDI $t0, $zero, 105 # i
SB $t0, 1($sp)
ADDI $t0, $zero, 32 # (space)
SB $t0, 2($sp)
```

```
ADDI $t0, $zero, 109 # m
SB $t0, 3($sp)
ADDI $t0, $zero, 111 # o
SB $t0, 4($sp)
ADDI $t0, $zero, 109 # m
SB $t0, 11($sp)
ADDI $t0, $zero, 0 # (null)
SB $t0, 12($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)

```
//L-M Miranda, #1
#include <iostream>
using namespace std;
int main()
{
  cout << "I wanna build\nSomething ";
  cout << "that's gonna";
  cout << "\nOutlive me" << endl;
}
```

Answer Key:

```
I wanna build
Something that's gonna
Outlive me
```

(b)

```
//L-M Miranda, #2
#include <iostream>
using namespace std;
int main()
{
  int count = 4;
  while (count > 0) {
    cout <<"\tWhat do you want, Burr?\n";
    count--;
  }
}
```

Answer Key:

```
    What do you want, Burr?
    What do you want, Burr?
    What do you want, Burr?
    What do you want, Burr?
```

(c)

```
//L-M Miranda, #3
#include <iostream>
using namespace std;
int main()
{
  cout << "If we lay a strong enough found";
  cout << "ation\nWe’ll pass it on to you, ";
  cout << "we’ll give the world to you\n";
  for (int i = 0; i < 2; i++) {
    if (i % 2 == 0)
      cout << "And ";
    else
      cout << "Yeah, ";
    cout << "you’ll blow us all away...\n";
    cout << "Someday, someday" << endl;
  }
}
```

Answer Key:

```
If we lay a strong enough foundation
We’ll pass it on to you, we’ll give the world to you
And you’ll blow us all away...
Someday, someday
Yeah, you’ll blow us all away
Someday, someday
```

10. (a) Write a **complete Python program** that simulates a random walk using the `turtle library`. Your turtle, at each iteration, should choose a random direction and walk forward 10 steps. The walk should continue as long as the turtle’s x and y coordinate values are within 100 of the starting point. **Answer Key:**

```
import turtle
import random

trey = turtle.Turtle()
trey.speed(10)

while (-100 < trey.xcor() < 100) and (-100 < trey.ycor() < 100):
  trey.forward(10)
  a = random.randrange(360)
  trey.right(a)
```

(b) Write a **complete C++ program** that prints out the yearly population, following the formula:

$$p(n+1) = 1.10 \cdot p(n) - \frac{.20 \cdot (p(n))^2}{20000}$$

where $p(n)$ is the population at year n. Assume that the starting population at year 0 is 600. Your program should print the year and population for the first 10 years.

Answer Key:

```
//Checks input for positive number
#include <iostream>
using namespace std;
int main()
{
  float p = 600;
  int year;
  for (year = 0; year < 10; year++) {
    cout << "Year:\t" << year << "population:\t" << p << "\n";
    p = 1.10*p - (.2*p*p)/20000;
  }
  return 0;
}
```