Final Exam, Version 4
CSci 127: Introduction to Computer Science
Hunter College, City University of New York

28 May 2019

Answer Key:

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

i.
```
s = "The%North**The%Mountain**The%Islands**The%Rock**The%Stormlands"
kingdoms = s.split('**')
print(s.count('**'))
print(s[-20:-16])
```

Answer Key:
```
4
Rock
```

ii.
```
kingdoms = s.split('**')
m = kingdoms[1]
words =  m.split('%')
print(words[1].upper())
```

Answer Key:
```
MOUNTAIN
```

iii.
```
for kingdom in kingdoms:
    word = kingdom.split('%')
    print(word[1])
```

Answer Key:
```
North
Mountain
Islands
Rock
```

```
Stormlands
```

(b) Consider the following shell commands:

```
$ ls
homework  p40.py  p41.py  p55.cpp  trees.csv
```

i. What is the output for:

```
$ cp p55.cpp prog55.cpp
$ ls
```

Answer Key:

```
homework  p40.py  p41.py  p55.cpp prog55.cpp  trees.csv
```

ii. What is the output for:

```
$ mv p*.* homework
$ ls | grep *.*
```

Answer Key:

```
trees.csv
```

iii. What is the output for:

```
$ pwd
/Users/yourlogin
$ echo "hello, you are in $PWD."
```

Answer Key:

```
hello, you are in /Users/yourlogin.
```

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

Answer Key:

	Decimal:	Hexadecimal:	Equal
a)	254	**FF**	*Equal*
b)	10	A	**Equal**
c)	**38**	25	*Equal*
d)	22	16	**Equal**
e)	20	**20**	*Equal*

(b) Given the function below

```
def decimalToOctString(decNum):
    octString = ""
    while decNum > 0:
        if decNum % 8 == 0:
            lead = '0'
        else:
            lead = str(decNum % 8)
        octString = lead + octString
        decNum = decNum // 8
    print(octString)
```

i. What is the output of `decimalToOctString(8)`

Answer Key:

```
10
```

ii. What is the output of `decimalToOctString(15)`

Answer Key:

```
17
```

What is the output of `decimalToOctString(64)`

Answer Key:

```
100
```

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

i.
```
in1 = False
in2 = True
out = not in1 or not in2
```

Answer Key:

```
out = True
```

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

Answer Key:

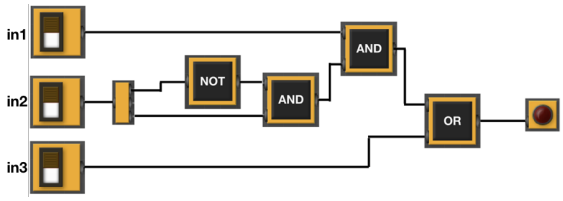
```
out = True
```

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

Answer Key:

```
out = True
```

iv.




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

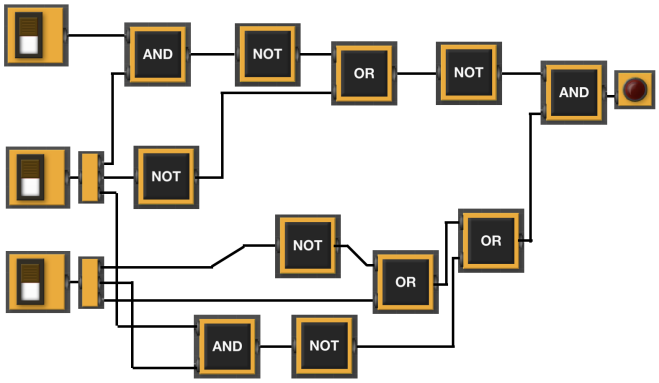
Answer Key:

```
out = False
```

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

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

Answer Key:



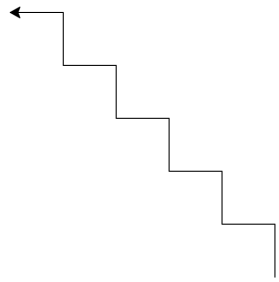
4. (a) Draw the output for the function calls:

```
import turtle
tammy = turtle.Turtle()

def mystery(tommy, n):
    tommy.left(180)
    for i in range(n):
        tommy.right(90)
        tommy.forward(50)
        tommy.left(90)
        tommy.forward(50)
```

i. `mystery(tammy, 5)`

Answer Key:



ii. `what are the formal parameters of mystery1?`
Answer Key: tommy,n

(b) Given the function definition:

```
def enigma(n):
   for i in range(n,0,-1):
      help(i)
      print()
```

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

i. What is the output for `enigma(6)`?

Answer Key:

0 1 0 1 0 1
1 0 1 0 1
0 1 0 1
1 0 1
0 1
1

5. Design an algorithm that prints out the number of 311 calls reporting "Illegal Parking" after a user-specified date from the NYC 311 calls OpenData. Specify the libraries, inputs and outputs for your algorithm and give the design in pseudocode.

Unique Key	Created Date	Closed Date	Agency	Agency Name	Complaint Type
42102569	04/01/2019 12:00:14 AM	04/01/2019 07:43:05 AM	NYPD	New York City Police Department	Noise - Residential
42101059	04/01/2019 12:00:21 AM	04/01/2019 06:12:11 AM	NYPD	New York City Police Department	Illegal Parking
42099515	04/01/2019 12:01:01 AM	04/01/2019 02:27:27 AM	NYPD	New York City Police Department	Blocked Driveway
42103744	04/01/2019 12:01:31 AM	04/01/2019 01:51:02 AM	NYPD	New York City Police Department	Noise - Residential
42102533	04/01/2019 12:01:50 AM	04/01/2019 12:24:02 AM	NYPD	New York City Police Department	Illegal Parking
42102278	04/01/2019 12:03:02 AM	04/01/2019 01:51:01 AM	NYPD	New York City Police Department	Noise - Residential
42098650	04/01/2019 12:03:11 AM	04/01/2019 05:33:50 PM	NYPD	New York City Police Department	Noise - Residential
42107429	04/01/2019 12:03:41 AM	04/03/2019 10:46:33 AM	HPD	Department of Housing Preservation	HEAT/HOT WATER
42110677	04/01/2019 12:04:37 AM	04/01/2019 12:04:37 AM	DOB	Department of Buildings	Building/Use
42103502	04/01/2019 12:04:38 AM	04/01/2019 07:05:09 AM	NYPD	New York City Police Department	Noise - Residential

Libraries:

Answer Key: pandas
Input:

Answer Key: The name of the CSV file and the year
Output:

Answer Key: The number of calls.

Process:

Answer Key:

(a) Ask user for file name and year.

(b) Open the file as a dataFrame.

(c) Select all the rows where `'Complaint Type'` is `'Illegal Parking'` and `'Created Date'` is after the date entered by the user.

(d) Print out the number of selected rows.

6. Fill in the Python program that will:
   - prompt the user for the name of the input file
   - prompt the user for the name of the output file
   - read the image from the input file into a data frame
   - compute the height and width of the image
   - extract the **right quarter** of the image and save it to the output file


```
#P6,V4: saves the right quarter of an image
#Import the libraries for storing and displaying images:
```

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

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

```
#Read image into a numpy array:
```

```
#Compute the height of the image
```

```
#Compute the width of the image
```

```
# Select right quarter and store in rightQuarterImg
```

```
#Save the right quarter image
```

Answer Key:

```
#P6,V4: saves the right quarter of an image

#Import the libraries for storing and displaying images:
import numpy as np
import matplotlib.pyplot as plt

#Prompt user for input file name:
inFileName = input('Enter input image: ')

#Prompt user for output file name:
outFileName = input('Enter ouput image: ')

#Read image into a numpy array:
img = plt.imread(inFileName)

#Compute the height of the image
height = img.shape[0]

#Compute the width of the image
width = img.shape[1]

# Select right quarter and store in rightQuarterImg
rightQuarterImg = img[ : , (width//4)*3 : ]

#Save the right quarter image
plt.imsave(outFileName, rightQuarterImg)
```

7. Complete the following program, based on the payroll dataset in the image below and the comments in the functions:

Fiscal Year	Agency Name	Agency Start Date	Work Location Borough	Title Description	Base Salary	Pay Basis	Regular Hours	OT Hours
2018	BOARD OF ELECTION	07/28/2014	MANHATTAN	TEMPORARY CLERK	13.79	per Hour	234.18	75.75
2018	BOARD OF ELECTION	02/28/2016	QUEENS	TEMPORARY CLERK	15	per Hour	1664.55	87
2018	BOARD OF ELECTION	03/13/2016	BRONX	FINANCIAL CLERK	19.79	per Hour	1638.88	66.25
2018	BOARD OF ELECTION	10/02/2017	BRONX	TEMPORARY CLERK	15	per Hour	1195.75	57.5
2018	BOARD OF ELECTION	10/31/2016	BRONX	TEMPORARY CLERK	15	per Hour	1339.38	60.75
2018	BOARD OF ELECTION	06/11/2012	BRONX	TEMPORARY CLERK	15	per Hour	1258.75	58.25

Answer Key:

```
import pandas as pd

def readDataFrame():
    inFile = input('Enter input file name: ')
    salaries = pd.read_csv(inFile)
    return(salaries)

def alterDataFrame(df):
    newColName = input('Enter the name of the new column: ')
    df[newColName] = df['Regular Hours'] + df['OT Hours']
    return(df, newColName)

def printColumnAverage(df, column):
    avg = df[column].mean()
    print(avg)

def main():
    df = readDataFrame()
    df2, newColName = alterDataFrame(df)
    printColumnAverage(df2, newColName)

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

8. (a) What are the values of register $s0 for the run of this MIPS program:

```
#Sample program that loops up to 50
ADDI $s0, $zero, 10 #set s0 to 10
ADDI $s1, $zero, 5 #use to increment counter, $s0
ADDI $s2, $zero, 50 #use to compare for branching
AGAIN: ADD $s0, $s0, $s1
BEQ $s0, $s2, DONE
J AGAIN
DONE: #To break out of the loop
```

Answer Key:

```
10
15
20
25
30
35
40
45
50
```

(b) Indicate what modifications are needed to the MIPS program (repeated below) so that it decrements by 10 all the way down to 0 (shade in the box for each line that needs to be changed and rewrite the instruction in the space below).

Answer Key:

```
#Sample program that loops up to 50
ADDI $s0, $zero, 0 #set s0 to 0
ADDI $s1, $zero, 10 #use to increment counter, $s0
ADDI $s2, $zero, 50 #use to compare for branching
AGAIN: ADD $s0, $s0, $s1
BEQ $s0, $s2, DONE
J AGAIN
DONE: #To break out of the loop
```

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

(a)

```
//Quote by George R.R. Martin, A Game of Thrones
#include <iostream>
using namespace std;
int main(){
  cout << "We are only human and";
  cout << "\nthe gods have fashioned ";
  cout << "us for love." << endl;
  cout << "That is our great glory\n";
  cout << "and our great tragedy.";
  return 0;
}
```

Answer Key:

```
We are only human and
the gods have fashioned us for love.
That is our great glory
and our great tragedy.
```

(b)

```
//More GOT
#include <iostream>
using namespace std;
int main(){
  int count = 0;
  while (count < 3) {
    cout <<"Laughter is... \n";
    count++;
  }
  cout << "\npoison to fear.";
  cout << endl;
  return 0;
}
```

Answer Key:

```
Laughter is...
Laughter is...
Laughter is...

poison to fear.
```

(c)

```
#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)
                cout << "*";
            else if( j % 2 == 0)
                cout << "O";
            else
                cout << "X";
        cout << endl;
    }
  return 0;
}
```

Answer Key:

```
OX*XO
OX*XO
OX*XO
OX*XO
OX*XO
```

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

```
#Python Loops, V4:
for i in range(50,0,-10):
    print('*', i, '*')
```

Answer Key:

```
//C++ Loop, V4
#include <iostream>
using namespace std;
int main()
{
    int i;
    for (i = 50; i > 0; i=i-10) {
        cout << "* " << i << " *" << endl;
    }
```

```
  return 0;
}
```

(b) Assume that sea level rises 0.75% each year. Write a **complete C++ program** that asks the user for the starting elevation and computes the number of years it will take until the coast is under water.

Answer Key:

```
#include <iostream>
using namespace std;
int main()
{
    cout << "Please enter the initial elevation in ft: ";
    double elevation = 0;
    double sea_level = 0;
    cin >> elevation;
    int years = 0;
    while(sea_level <= elevation)
    {
        std::cout << sea_level << std::endl;
        sea_level = sea_level + 0.0075;
        years++;
    }

    cout << "It will take " << years << " years until the coast is under water.\n";


    return 0;
}
```