

FINAL EXAM, VERSION 1
CSci 127: Introduction to Computer Science
Hunter College, City University of New York

13 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:

Name:

EmpID:

CSci 127 Final, F18, V1

(Empty page for scratch work)

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

```
s = "avram,henriette;dolciani,mary;rees,mina"
i. 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[-4:])
```

Output:

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

Output:

- (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"
```

Output:

- ii. What is the output for:

```
$ ls *.html | wc -l
```

Output:

- iii. What is the output for:

```
$ ls -l | grep "Jul" | wc -l
```

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)	1	1	1	<i>All Equal</i>
b)	11	11	11	<i>All Equal</i>
c)	11111	29	1C	<i>All Equal</i>
d)	100000	34	20	<i>All Equal</i>
e)	11111111	255	FF	<i>All Equal</i>

- (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]):

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

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

in1 = True

i. in2 = False

out = in1 and in2

out =

in1 = False

ii. in2 = False

out = not in1 and (in2 or not in1)

out =

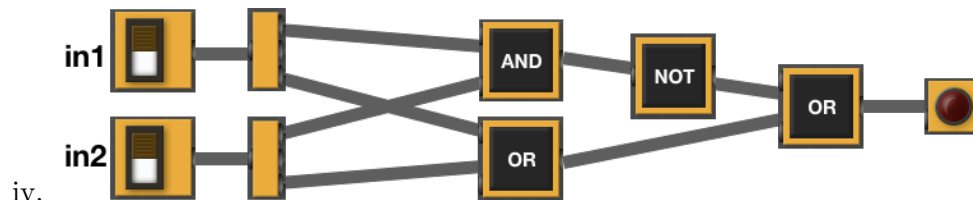
in1 = True

iii. in2 = False and not in1

in3 = in1 and in2

out = in1 and not in3

out =



in1 = True

in2 = False

out =

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

$((in1 \text{ or } in2) \text{ and } (in1 \text{ or } in3)) \text{ or } (\text{not } (in2 \text{ and } in3))$

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()`:
- ii. What are the formal parameters for `startV1()`:
- iii. What does `startV1(30)` return:

- (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]
```

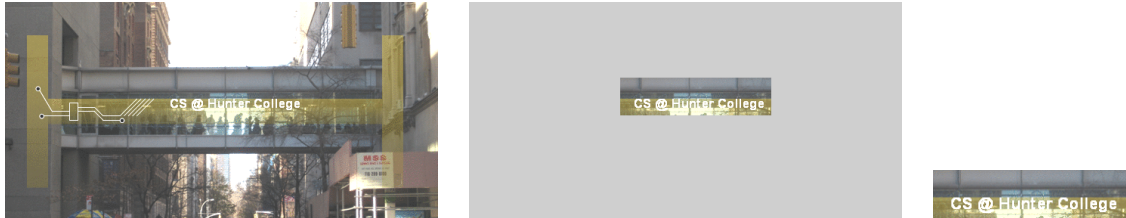
- i. What is the output for `sorted([9,8,1,3])`?

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

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

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

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:

Output:

Process:

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:
```


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.

```
#Intro Programming Lab: A program with herd of turtles
```

```
import turtle
```

```
def getInput():
```

```
    """Prompts & returns the number of turtles the user entered."""
```

```
def setUpTurtles(n):
```

```
    """Creates a list of n turtles and returns the list."""
```

```
    tList = []
```

```
    #Create turtles:
```

```
    return tList
```

```
def moveForward(tList):
```

```
    """Move each turtle in the list forward 30 steps."""
```

```
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
```

Output:

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

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

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

Output:

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

Output:

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

Output:

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

```
#Python Loop, V1:  
for i in range(5,101,5):  
    print(i)
```

- (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 $\frac{r f}{5}$ are eaten by foxes. The fox population decreases by 5% due to old age but increases in proportion to the food supply, $\frac{r f}{10}$. 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.