

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

12 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.
- 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:

i.

```
mes = "Get your education"
print(mes.upper())
```

Output:

ii.

```
mess = "daoanaata afaoaragaeat"
mList = mess.split('a')
decode = "".join(mList)
print(decode)
```

Output:

iii.

```
messa = "From Whence You Came."
print(messa.lower())
```

Output:

iv.

```
prefix = ["And I c", "And we c", \
          "That w"]
for p in prefix:
    s = p+"ould be enough"
    print(s)
```

Output:

- (b) Consider the following shell commands:

```
$ ls -l
drwxr-xr-x  32 stjohn  staff      1088 May 18  2018 drafts/
-rw-r--r--@  1 stjohn  staff     246352 May 15  2018 examMapFinal.pdf
-rw-r--r--@  1 stjohn  staff     571936 May 22  2018 examMapFinalCropped.jpg
-rwxrwxrwx@  1 stjohn  staff    1136855 May 14  2018 finalS18V1.pdf*
-rwxrwxrwx@  1 stjohn  staff    1125569 May 14  2018 finalS18V2.pdf*
drwxr-xr-x  21 stjohn  staff       714 May 23  2018 sign-in/
drwxr-xr-x   7 stjohn  staff       238 May 18  2018 submissions/
```

- i. What is the output for:

Output:

```
$ ls *.pdf
```

- ii. What is the output for:

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):

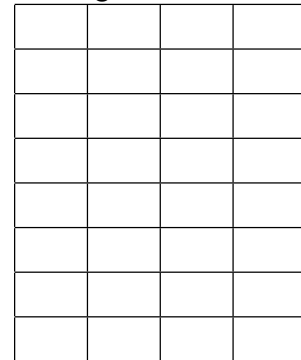
	Binary:	Decimal:	Hexadecimal:	All Equal
a)	10	10	10	<i>All Equal</i>
b)	1000	15	F	<i>All Equal</i>
c)	10001	17	11	<i>All Equal</i>
d)	100000	33	20	<i>All Equal</i>
e)	1111111	250	FF	<i>All Equal</i>

- (b) After the code is run, in the grid below:

- Shade in the entries that are assigned the color black,
- Mark with an ‘P’ the entries that are assigned purple, and
- Leave empty entries that are assigned white.

```
import matplotlib.pyplot as plt
import numpy as np
examImg = np.ones((8,4,3))
examImg[0,0,1] = 0
examImg[0,2,1] = 0
examImg[1,1,1] = 0
examImg[1,3,1] = 0
examImg[2:4, :, :] = 0
examImg[4::2, :, 1] = 0
plt.imshow(examImg)
plt.show()
```

examImg:



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

in1 = True

i. in2 = False

out = in1 or in2

out =

in1 = False

ii. in2 = True

out = not in1 and (in2 and not in1)

out =

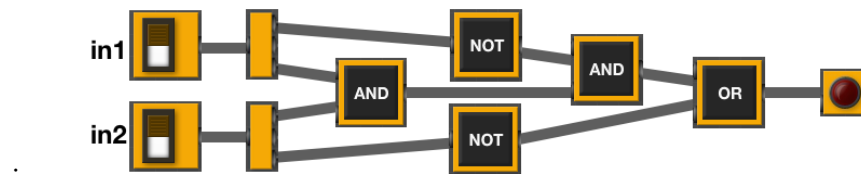
in1 = True

iii. in2 = False or not in1

in3 = in1 and in2

out = in1 and not in3

out =



iv.

in1 = True

in2 = False

out =

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

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

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

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

def ramble(t, len, isNested):
    if len >= 10:
        for i in range(3):
            t.forward(len)
            t.left(120)
            if isNested:
                ramble(t, len-10, isNested)
```

i. `ramble(trey,20,False)` ii. `ramble(trey,20,True)`

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

```
def searchMe(list1, element):
    if len(list1) == 0:
        return -1
    else:
        mid = len(list1)//2
        if (element == list1[mid]):
            return mid
        else:
            if element > list1[mid]:
                return searchMe(list1[mid+1:], element)
            else:
                return searchMe(list1[:mid], element)
```

i. `searchMe([1,3,5], 3)`

Return:

ii. `searchMe([1,3,5,7,9], 3)`

Return:

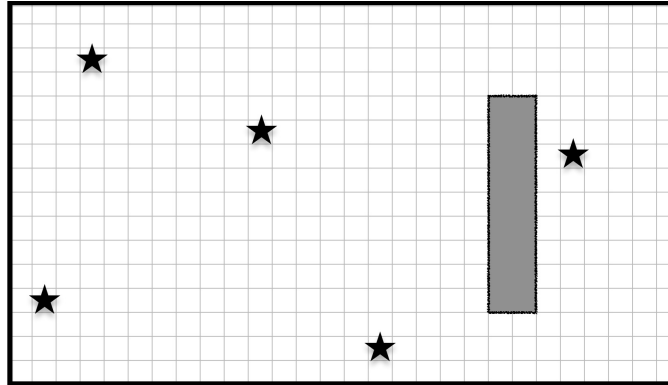
iii. `searchMe([1,3,5,7,9], 2)`

Return:

iv. `searchMe([1,3,5,7,9], 11)`

Return:

5. Design an algorithm that uses turtles to explore a space and visits all 5 stars. The star locations change each time. Specify the inputs and outputs for your algorithm and give the design in pseudocode.



Your turtle has an additional function, `sensor()`, that returns a value depending on what is in the square in front of you: 0 if there is wall, 1 if there is a star, and 2 otherwise.

• **Input:**

• **Output:**

• **Process:**

6. Using `pandas`, write a **complete Python program** that asks the user for a recipe (in comma separated value (CSV) format), reads in the corresponding CSV file and prints out quantities and ingredients needed to make a double batch. Assume that the CSV files have the columns: "Amount", "Measurement", and "Ingredient".

For example if the CSV file, `meringues.csv`, contained:

Amount	Measurement	Ingredient
150	grams	chocolate chips
4	whites of	eggs
.25	teaspoon	vanilla
.25	teaspoon	cream of tartar

A sample run of your program would be:

```
Enter recipe name: meringues.csv
  Amount Measurement      Ingredient
0   300.0      grams chocolate chips
1     8.0 whites of           eggs
2     0.5   teaspoon          vanilla
3     0.5   teaspoon  cream of tartar
```

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

- `setUp()`: sets up a graphics window and turtle
- `drawDecagon()`: draws a decagon (10-sided figure), and
- `conclusion()`: then prints a closing message and closes the graphics window when mouse is clicked

```
import turtle
def setUp():
    """
    Returns a new graphics window and turtle.
    """
```

```
def drawDecagon(t):
    """
    Takes a turtle as input.
    Draws a decagon (10-sided figure).
    """
```

```
def conclusion(w):
    """
    Takes a window as input.
    Prints a closing message and closes graphics window when clicked.
    """
```

```
def main():
    w,t = setUp()    #sets up a graphics window and turtle
    drawDecagon(t)  #draws a decagon using the turtle
    conclusion(w)   #prints goodbye and closes window on click
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, 65 # Start $t0 at 65 (A)
ADDI $s2, $zero, 70 # Use to test when you reach 70 (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 == 70
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) What modifications are needed to the MIPS program above so that it prints out the first 10 upper case letters: ABCDEFGHIJ ?

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

```
//Dr. Seuss, Places You'll Go:
#include <iostream>
using namespace std;
int main()
{
    cout << "And will you succeed?" << endl;
(a)  cout << "Yes! You will, ";
    cout << "indeed!\n(98 and 3/4 percent";
    cout << " guaranteed.) "<< endl;
}
```

Output:

```
//More Dr. Seuss, Cat in the Hat:
#include <iostream>
using namespace std;
int main()
{
    int count = 0;
    while (count < 3) {
(b)  cout << endl << "Look at me";
        count++;
    }
    cout << "NOW\nIt is fun to have fun\n";
    cout << "But you have to know how.";
}
```

Output:

```
//Stars and more stars
#include <iostream>
using namespace std;
int main()
{
    int i, j;
(c)  for (i = 5; i > 0; i--)
    {
        for (j = 1; j <= i; j++)
            cout << "*";
        cout << endl;
    }
}
```

Output:

10. (a) Write a complete **Python program** that prints the numbers from 1 to 100. But for multiples of three print “Fizz” instead of the number and for the multiples of five print “Buzz”. For numbers which are multiples of both three and five print “FizzBuzz”.

- (b) Write a **complete C++ program** that repeatedly prompts the user for the year they were born until they enter a number that is 2018 or smaller. Your program should print out the final number the user entered: