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

12 December 2018

Answer Key:

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

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

    Answer Key:

    ```
    GET YOUR EDUCATION
    ```

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

    Answer Key:

    ```
    dont forget
    ```

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

    Answer Key:

    ```
    from whence you came.
    ```

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

Answer Key:

```
And I could be enough
And we could be enough
That would be enough
```

(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:
```
$ ls *.pdf
```

Answer Key:

```
examMapFinal.pdf
finalS18V1.pdf*
finalS18V2.pdf*
```

  ii. What is the output for:
```
$ ls final* | wc -w
```

Answer Key:

```
2
```

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):

Answer Key:

		Binary:	Decimal:	Hexadecimal:	All Eq
	a)	10	10	**10**	*All Eq*
	b)	**10000– Typo in printed exam**	15	F	*All Eq*
	c)	10001	17	11	**All Eq**
	d)	100000	**33**	20	*All Eq*
	e)	1111111	250	**FF**	*All Eq*

(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()
```

Answer Key:
examImg:

P		P	
	P		P
■	■	■	■
■	■	■	■
P	P	P	P
P	P	P	P

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

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

Answer Key:
```
out = True
```

```
    in1 = False
ii. in2 = True
    out = not in1 and (in2 and not in1)
```
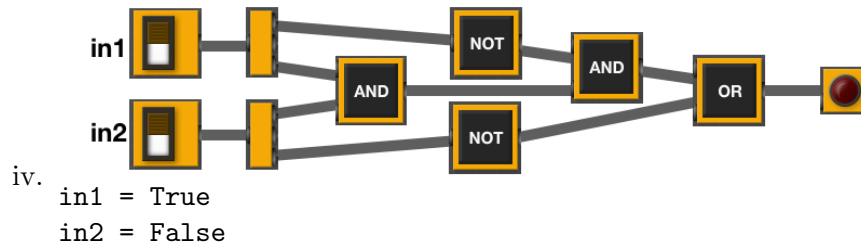
Answer Key:

```
out = False
```

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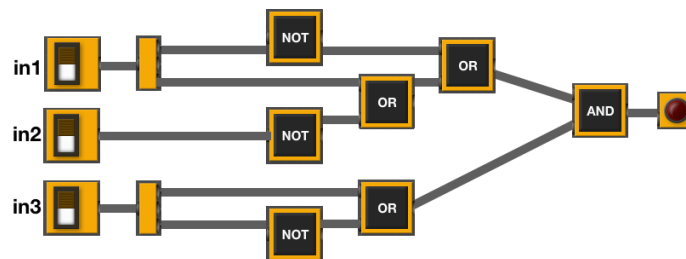
Answer Key:

```
out = True
```

iv.


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

Answer Key:

```
out = True
```

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

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


Answer Key:

4. (a) Draw the output for the function calls:
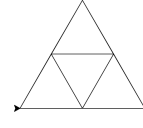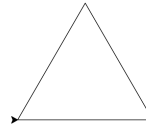
```
import turtle
trey = turtle.Turtle()

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)`

Answer Key: **Answer Key:**

(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)`

   Answer Key: 1

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

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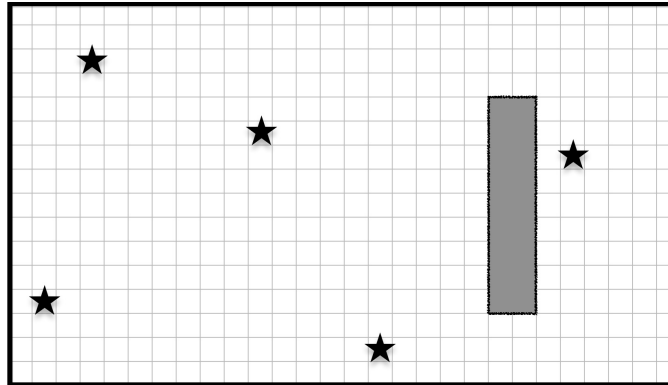
iii. `searchMe([1,3,5,7,9],2)`

   Answer Key: -1

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

   Answer Key: -1

5. Design an algorithm that uses turtles to explores 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:**

  Answer Key: A map/grid of the world.

- **Output:**

  Answer Key: Time taken and/or location of the 5 stars.

- **Process:**

  Answer Key:
  (a) Set up a numpy array, map, with −1 everywhere.
  (b) Set numStars = 0
  (c) While numStars < 5:
      i. Move forward.
      ii. If wall, mark 0 in map, randomly turn left or right.
      iii. If star, mark 1 in map and add 1 to numStars.
      iv. Otherwise, mark 2 in map that it?s an empty square.
  (d) Return the map.

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
```

Answer Key:

```python
import pandas as pd

fileName = input('Enter recipe name: ')
recipe = pd.read_csv(fileName)

recipe['Amount'] = 2*recipe['Amount']
print(recipe)
```

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

Answer Key:

```python
import turtle

def setUp():
    trey = turtle.Turtle()
    win = turtle.Screen()
    return(win,trey)

def drawDecagon(t):
    for i in range(10):
```

```
        t.forward(100)
        t.right(360/10)

def conclusion(w):
    print("Goodbye!")
    w.exitonclick()

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 $to 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
```

Answer Key:

ABCDE

(b) What modifications are needed to the MIPS program above so that it prints out the first 10 upper case letters: ABCDEFGHIJ ?

Answer Key: Need to change:

- the first line to have space for 11 characters (AB...J and the null to terminate).
- the corresponding line to allow 11 characters to print (i.e. `ADDI $sp, $sp, -11`).
- the value of $s2 to be 75 (K).

The resulting program:

```
#Loop through first 10 letters:
ADDI $sp, $sp, -11   # Set up stack
ADDI $t0, $zero, 65 # Start $t0 at 65 (A)
ADDI $s2, $zero, 76 # Use to test when you reach 75 (K)
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 == 75
J SETUP              # If not, jump back to SETUP for loop
DONE: ADDI $t0, $zero, 0 # Null to end the string
SB $t0, 0($sp)       # Add null to stack
ADDI $sp, $sp, -11   # 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)
```
//Dr. Seuss, Places You'll Go:
#include <iostream>
using namespace std;
int main()
{
  cout << "And will you succeed?" << endl;
  cout << "Yes! You will, ";
  cout << "indeed!\n(98 and 3/4 percent";
  cout << " guaranteed.) "<< endl;
}
```

Answer Key:

```
And will you succeed?
Yes! You will, indeed!
(98 and 3/4 percent guaranteed.)
```

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

(b)

Answer Key:

```
Look at me
Look at me
Look at me NOW
It is fun to have fun
But you have to know how.
```

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

(c)

Answer Key:

```
*****
****
***
**
*
```

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".

Answer Key: Many ways to do this program. Here's one:

```
#FizzBuzz program-- from Lecture 4
for i in range(1,101):
        if i%3 != 0 and i%5 != 0:
                print(i, end="")
        if i%3 == 0:
                print("Fizz", end="")
        if i%5 == 0:
                print("Buzz", end="")
        print()
```

(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:

Answer Key:

```
//Checks input for year born
#include <iostream>
using namespace std;
int main()
{
  int year;
  cout << "Please enter the year you were born: ";
  cin >> year;
  while (year > 2018) {
    cout << "You entered a year in the future.\n";
    cout << "Please enter the year you were born: ";
    cin >> year;
  }
  cout << "Year your were born: " << year;
  return 0;
}
```