

Answer Key:

FINAL EXAM, VERSION 3
CSci 127: Introduction to Computer Science
Hunter College, City University of New York

16 December 2019

1. (a) What will the following Python code print:
- ```
pioneers = "Jones-Karen Spark;Jobs-Steve;Gates-Bill"
i. print(pioneers[-4:],pioneers[-10:-5])
 print(pioneers.count('-'))
```

**Answer Key:**

Bill Gates  
3

- ```
names = pioneers.split(';')
ii. m = names[1]
    print(m[:4])
```

Answer Key:

Jobs

- ```
for n in names:
iii. print(n.split('-')[0].upper())
```

**Answer Key:**

JONES  
JOBS  
GATES

- (b) Consider the following shell commands:

```
$ ls
snow.png p30.py p40.py tickets.png
```

- i. What is the output for:  
\$ ls \*png

**Answer Key:**

```
snow.png tickets.png
```

- ii. What is the output for:  
`$ ls | grep py | wc -l`

**Answer Key:**

2

- iii. What is the output for:  
`$ mkdir new`  
`$ cd new`  
`$ touch stars.png`  
`$ ls`

**Answer Key:**

stars.png

2. (a) Consider the code:

**Answer Key:**

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

- i. After the command: `thomasH.color("#00DD00")`, what color is `thomasH`?  
 black       green       white       gray       purple
- ii. After the command: `thomasH.color("#FFFFFF")`, what color is `thomasH`?  
 black       green       white       gray       purple
- iii. Fill in the code below to change `thomasH` to be the brightest red:  
`thomasH.color("# 

|   |   |   |   |   |   |
|---|---|---|---|---|---|
| F | F | 0 | 0 | 0 | 0 |
|---|---|---|---|---|---|

 ")`
- iv. Fill in the code below to change `thomasH` to be the color black:  
`thomasH.color("# 

|   |   |   |   |   |   |
|---|---|---|---|---|---|
| 0 | 0 | 0 | 0 | 0 | 0 |
|---|---|---|---|---|---|

 ")`
- (b) Fill in the code to produce the output on the right:

- i. **Answer Key:** `for i in range( 9 ):  
print(i, end=" ")`

**Output:**

```
0 1 2 3 4 5 6 7 8
```

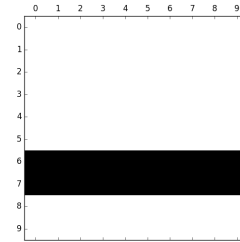
- ii. **Answer Key:** `for j in range( -1, 4, 1 ):  
print(i, end=" ")`

**Output:**

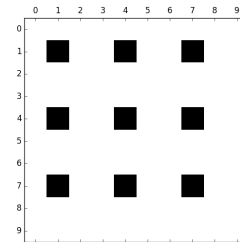
```
-1 0 1 2 3
```

**Answer Key:**

```
import numpy as np
import matplotlib.pyplot as plt
iii. im = np.ones((10,10,3))
 im[6:7, :, :] = 0
 plt.matshow(im)
 plt.show()
```

**Output:****Answer Key:**

```
import numpy as np
import matplotlib.pyplot as plt
iv. im = np.ones((10,10,3))
 im[1:3, 1:3, :] = 0
 plt.matshow(im)
 plt.show()
```

**Output:**

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

```
in1 = False
```

i. `in2 = True`

```
out = in1 or in2
```

**Answer Key:**

```
out = True
```

```
in1 = True
```

ii. `in2 = True`

```
out = not in1 or (in2 and not in2)
```

**Answer Key:**

```
out = False
```

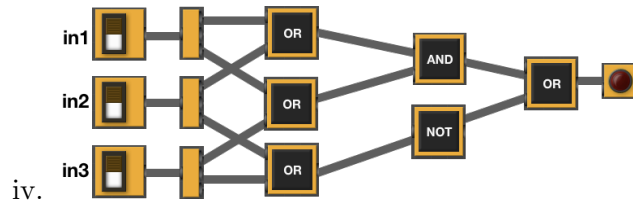
```
in1 = True
```

iii. `in2 = True or not in1`

```
in3 = in1 or in2
out = in1 and not in3
```

**Answer Key:**

```
out = False
```



in1 = True  
in2 = False  
in3 = False

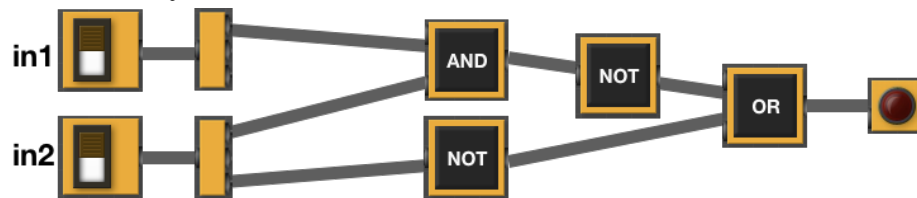
**Answer Key:**

out = True

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

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

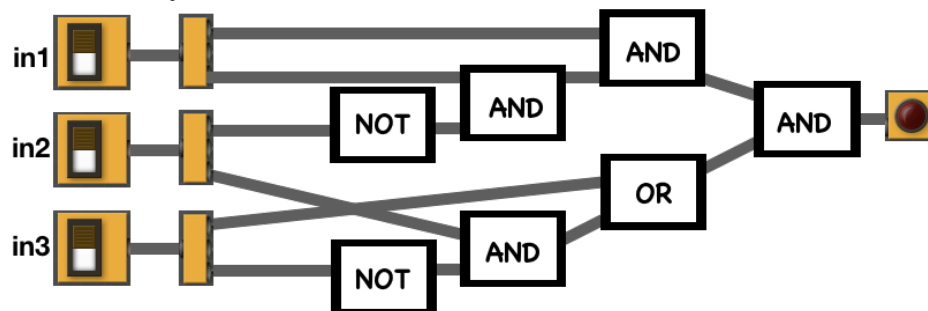
**Answer Key:**



(c) Fill in the circuit that implements the logical expression:

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

**Answer Key:**



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

i. `ramble(tyler,4,True)`

**Answer Key:**

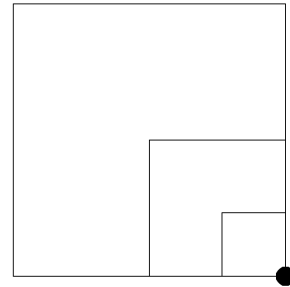
```
import turtle
tyler = turtle.Turtle()
tyler.shape('circle')

def ramble(tori, dist, repeat):
 if dist > 5:
 for i in range(4):
 tori.left(90)
 tori.forward(dist*10)
 ramble(tori,dist//2,repeat)
 elif repeat:
 for i in range(dist):
 tori.forward(20)
 tori.stamp()
 else:
 tori.stamp()
```



ii. `ramble(tyler,30,False)`

**Answer Key:**



(b) What are the formal parameters for `ramble()`:

**Answer Key:** `tori, dist, repeat`

(c) If you call `ramble(tyler,4,True)`, which branches of the function are tested: **Answer**

**Key:**

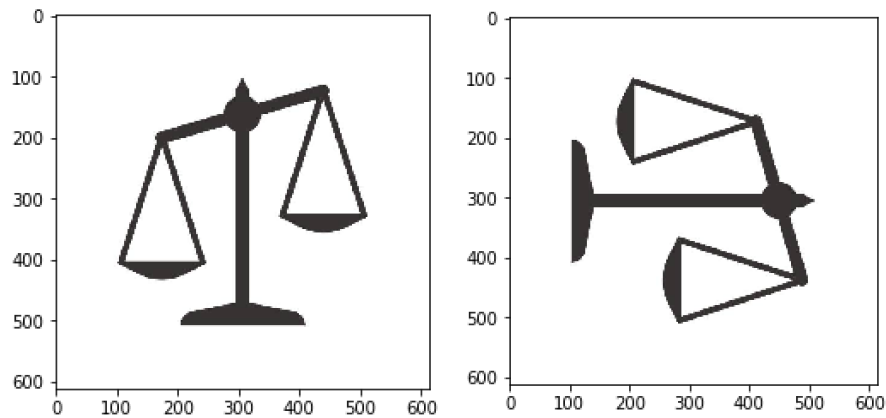
- the `if`-clause only,
- the `elif`-clause only,
- the `else`-clause only,
- `if`-clause and the `else`-clause, or
- all the clauses are visited from this invocation (call).

(d) If you call `ramble(tyler,30,False)`, which branches of the function are tested: **Answer**

**Key:**

- the `if`-clause only,
- the `elif`-clause only,
- the `else`-clause only,
- `if`-clause and the `else`-clause, or
- all the clauses are visited from this invocation (`call`).

5. Design an algorithm that rotates an image by 90 degrees to the right. For simplicity, you may assume a square image (i.e. same height and length)



**Libraries:**

**Answer Key:** `matplotlib.pyplot` and `numpy`

**Input:**

**Answer Key:** The name of the image file

**Output:**

**Answer Key:** The rotated image

**Process (as a list of steps):**

**Answer Key:**

- (a) Ask user for image file name
- (b) Read the image in a numpy array, call it `img`
- (c) Create a new numpy array with same dimensions, call it `img2`

- (d) Copy the first row of `img` into the last column of `img2`, such that `img[0,0,:] == img2[0,n,:]`, `img[0,1,:] == img2[1,n,:]`, ... , `img[0,n,:] == img2[n,n,:]`
- (e) Repeat analogous process to copy the second row of `img` into the second-to-last column of `img2`, third row of `img` into third-to-last column of `img2`, and so on for all rows in `img`
- (f) Save `img2`
6. Given the FiveThirtyEight dataset containing data on nearly 3 million tweets sent from Twitter handles connected to the Internet Research Agency, a Russian “troll factory”, a snapshot given in the image below:

| author | content                                             | region  | language | publish_date     | harvested_date   | following | followers | updates |
|--------|-----------------------------------------------------|---------|----------|------------------|------------------|-----------|-----------|---------|
| 10_GOP | "We have a sitting Democrat US Senator on trial     | Unknown | English  | 10/1/2017 19:58  | 10/1/2017 19:59  | 1052      | 9636      | 253     |
| 10_GOP | Marshawn Lynch arrives to game in anti-Trump s      | Unknown | English  | 10/1/2017 22:43  | 10/1/2017 22:43  | 1054      | 9637      | 254     |
| 10_GOP | JUST IN: President Trump dedicates Presidents       | Unknown | English  | 10/1/2017 23:52  | 10/1/2017 23:52  | 1062      | 9642      | 256     |
| 10_GOP | Dan Bongino: "Nobody trolls liberals better than    | Unknown | English  | 10/1/2017 2:47   | 10/1/2017 2:47   | 1050      | 9644      | 247     |
| 10_GOP | '@SenatorMenendez @CarmenYulinCruz Doesn'           | Unknown | English  | 10/1/2017 2:52   | 10/1/2017 2:53   | 1050      | 9644      | 249     |
| 10_GOP | As much as I hate promoting CNN article, here t     | Unknown | English  | 10/1/2017 3:47   | 10/1/2017 3:47   | 1050      | 9646      | 250     |
| 10_GOP | After the 'genocide' remark from San Juan Mayc      | Unknown | English  | 10/1/2017 3:51   | 10/1/2017 3:51   | 1050      | 9646      | 251     |
| 10_GOP | Sarah Sanders destroys NBC reporter: "Trump n       | Unknown | English  | 10/10/2017 20:57 | 10/10/2017 20:57 | 1066      | 10319     | 301     |
| 10_GOP | Hi @MichelleObama, remember when you praise         | Unknown | English  | 10/10/2017 22:06 | 10/10/2017 22:06 | 1066      | 10320     | 302     |
| 10_GOP | Wow! Even CNN is slamming the Obamas for sil        | Unknown | English  | 10/10/2017 22:17 | 10/10/2017 22:17 | 1066      | 10322     | 303     |
| 10_GOP | First lady Melania Trump visits infant opioid treat | Unknown | English  | 10/10/2017 23:42 | 10/10/2017 23:42 | 1068      | 10328     | 304     |
| 10_GOP | "It took Hillary abt 5 minutes to blame NRA for n   | Unknown | English  | 10/11/2017 20:26 | 10/11/2017 20:27 | 1070      | 10358     | 308     |

Fill in the Python program below:

### Answer Key:

```
#P6,V3: extracts dates with highest number of troll tweets

#Import the libraries for data frames and plotting data:
import pandas as pd
import matplotlib.pyplot as plt

#Prompt user for input file name:
csvFile = input('Enter CSV file name: ')

#Read input data into data frame:
trolls = pd.read_csv(csvFile)

#split date into date and time columns
trolls[['pub_date', 'pub_time']] = trolls.publish_date.str.split(expand=True)

#Count the number of tweets for each date:
trollDates = trolls["pub_date"].value_counts()

#Print the top 5 dates with most troll tweets
print(trollDates[:5])
```

```
#Generate a bar plot of the top 5 dates with largest number of troll tweets
trollDates.plot.bar()
plt.show()
```

7. Write a **complete Python program** that prompts the user for the name of an .png (image) file and prints the fraction of pixels that are very light. A pixel is very light if the red, green, and blue values are **all** over 90%.

**Answer Key:**

```
#Import the packages for images and arrays:
import matplotlib.pyplot as plt
import numpy as np
#Ask user for image name and read into img:
inImg = input('Enter input image: ')
img = plt.imread(inImg)
#Get height and width:
height = img.shape[0]
width = img.shape[1]
#Initialize counter:
count = 0
#Loop through all the pixels:
for row in range(height):
 for col in range(width):
 #Check if each pixel is very light and update count:
 if (img[row,col,0] > .9) and (img[row,col,1] > .9) and (img[row,col,2] > .9):
 count = count + 1
#Compute and print fraction:
frac = count/(height*width)
print('Fraction light is', frac)
```

8. (a) What is printed by the MIPS program below:

**Answer Key:**

!!!!!

- (b) Modify the program to print out 99 copies of the character '!'. Shade in the box for each line that needs to be changed and rewrite the instruction below.

**Answer Key:**

```
#Loop through characters
ADDI $sp, $sp, -100 # Set up stack
ADDI $s3, $zero, 1 # Store 1 in a registrar
ADDI $t0, $zero, 33 # Set $t0 at 33 (!)
```



```

ADDI $s2, $zero, 99 # Use to test when you reach 100
SETUP: SB $t0, 0($sp) # Next letter in $t0
ADDI $sp, $sp, 1 # Increment the stack
SUB $s2, $s2, $s3 # Decrease the counter by 1
BEQ $s2, $zero, DONE # Jump to done if $s0 == 0
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, -100 # 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?

```

//Quote by Bill Gates
#include <iostream>
using namespace std;
int main()
{
(a) cout<<"Weve got to put\na ";
 cout<<"lot of money into \nchanging";
 cout<<" behavior."<<endl<<"B.G.";
 return 0;
}

```

### Answer Key:

```

Weve got to put
a lot of money into
changing behavior.
#include <iostream>
using namespace std;
int main()
{
 double num = 0;
 double weight = 0;
 while (weight < 100) {
(b) cout <<"Please enter weight\n";
 cin >> weight;
 num += weight;
 }
 cout << num << endl;
 return 0;
}

```

**Answer Key:**

Please enter weight  
 Please enter weight  
 Please enter weight

```

#include <iostream>
using namespace std;
int main(){
 int i, j;
 for (i = 1; i <= 5; i++){
 for (j = 0; j < i; j++){
 if(j % 2 == 0)
(c) cout << "0";
 else
 cout << "X";
 }
 cout << endl;
 }
 return 0;
}

```

**Answer Key:**

0  
 0X  
 0X0  
 0X0X  
 0X0X0

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

```

#Python Loops, V3
for i in range(0,15,3):
 print(i, '*', i)

```

**Answer Key:**

```

//C++ Loop, V3
#include <iostream>
using namespace std;
int main()
{
 for(int i=0; i<15; i+=3)
 cout << i << " * " << i << endl;
 return 0;
}

```

- (b) The number of Facebook monthly active users grew from ~500 million in 2010 to ~2500 million (2.5 billion) in 2019. The average annual growth rate can then be estimated as

$$\text{avgGrowth} = \frac{\% \text{growth}}{\text{number-of-years}} = \frac{100 \cdot \frac{2500-500}{500}}{2019 - 2010} = 44.4\%$$

We can thus estimate an average annual growth: **avgGrowth = 44.4%**

Write a **complete C++ program** that asks the user for a year greater than 2010 (assume user complies) and prints the estimated number (in millions) of monthly active Facebook users in that year.

**Answer Key:**

```
//Facebook monthly active users V3
#include <iostream>
using namespace std;
int main()
{
 double past = 500;
 double avgGrowth = past * .444;
 int year = 0;

 cout << "Please enter a year between 2010 and 2019 : ";
 cin >> year;

 double users = (past + (avgGrowth * (year-2010)))/12;

 cout << "The number of monthly active Facebook users in ";
 cout << year << " is approximately " << users << " millions" << endl;

 return 0;
}
```