Row:	Seat:

Final Exam, Version 3
CSci 127: Introduction to Computer Science
Hunter College, City University of New York

16 December 2019

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, phone, earbuds, 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:								

ASCII TABLE

Decimal	Hex	Char	Decimal	Hex	Char	Decimal	Hex	Char	Decimal	Hex	Char
0	0	*[NULL]*	32	20	*[SPACE]*	64	40	@	96	60	`
1	1	*[START OF HEADING]*	33	21	!	65	41	A	97	61	a
2	2	*[START OF TEXT]*	34	22	"	66	42	B	98	62	b
3	3	*[END OF TEXT]*	35	23	#	67	43	C	99	63	c
4	4	*[END OF TRANSMISSION]*	36	24	$	68	44	D	100	64	d
5	5	*[ENQUIRY]*	37	25	%	69	45	E	101	65	e
6	6	*[ACKNOWLEDGE]*	38	26	&	70	46	F	102	66	f
7	7	*[BELL]*	39	27	'	71	47	G	103	67	g
8	8	*[BACKSPACE]*	40	28	(	72	48	H	104	68	h
9	9	*[HORIZONTAL TAB]*	41	29	)	73	49	I	105	69	i
10	A	*[LINE FEED]*	42	2A	*	74	4A	J	106	6A	j
11	B	*[VERTICAL TAB]*	43	2B	+	75	4B	K	107	6B	k
12	C	*[FORM FEED]*	44	2C	,	76	4C	L	108	6C	l
13	D	*[CARRIAGE RETURN]*	45	2D	-	77	4D	M	109	6D	m
14	E	*[SHIFT OUT]*	46	2E	.	78	4E	N	110	6E	n
15	F	*[SHIFT IN]*	47	2F	/	79	4F	O	111	6F	o
16	10	*[DATA LINK ESCAPE]*	48	30	0	80	50	P	112	70	p
17	11	*[DEVICE CONTROL 1]*	49	31	1	81	51	Q	113	71	q
18	12	*[DEVICE CONTROL 2]*	50	32	2	82	52	R	114	72	r
19	13	*[DEVICE CONTROL 3]*	51	33	3	83	53	S	115	73	s
20	14	*[DEVICE CONTROL 4]*	52	34	4	84	54	T	116	74	t
21	15	*[NEGATIVE ACKNOWLEDGE]*	53	35	5	85	55	U	117	75	u
22	16	*[SYNCHRONOUS IDLE]*	54	36	6	86	56	V	118	76	v
23	17	*[ENG OF TRANS. BLOCK]*	55	37	7	87	57	W	119	77	w
24	18	*[CANCEL]*	56	38	8	88	58	X	120	78	x
25	19	*[END OF MEDIUM]*	57	39	9	89	59	Y	121	79	y
26	1A	*[SUBSTITUTE]*	58	3A	:	90	5A	Z	122	7A	z
27	1B	*[ESCAPE]*	59	3B	;	91	5B	[	123	7B	{
28	1C	*[FILE SEPARATOR]*	60	3C	<	92	5C	\	124	7C	|
29	1D	*[GROUP SEPARATOR]*	61	3D	=	93	5D	]	125	7D	}
30	1E	*[RECORD SEPARATOR]*	62	3E	>	94	5E	^	126	7E	~
31	1F	*[UNIT SEPARATOR]*	63	3F	?	95	5F	_	127	7F	*[DEL]*

(Image from wikipedia commons)

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('-'))
```

Output:

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

Output:

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

Output:

(b) Consider the following shell commands:

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

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

Output:

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

Output:

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

Output:

2. (a) Consider the code:

```
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("#` ☐☐☐☐☐☐ `")`

iv. Fill in the code below to change `thomasH` to be the color black:

`thomasH.color("#` ☐☐☐☐☐☐ `")`

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

i.
```
for i in range( ☐ ):
    print(i, end=" ")
```

Output:

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

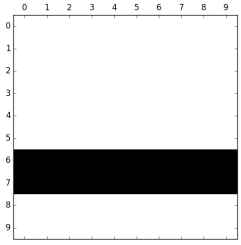
ii.
```
for j in range( ☐ , ☐ , ☐ ):
    print(i, end=" ")
```

Output:

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

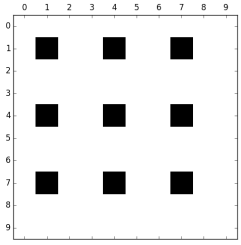
iii.
```
import numpy as np
import matplotlib.pyplot as plt
im = np.ones( (10,10,3) )
im[☐:7,:,:]  = 0
plt.matshow(im)
plt.show()
```

Output:



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

Output:



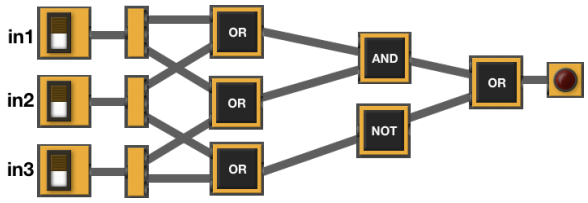
3. (a) What is the value (True/False):

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

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

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

iv.



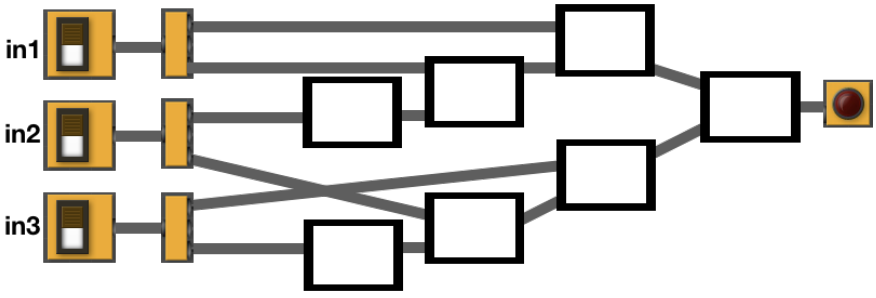

```
in1 = True
in2 = False
in3 = False
```
☐ True ☐ False

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

```
(not (in1 and in2) and (not in2))
```

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

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

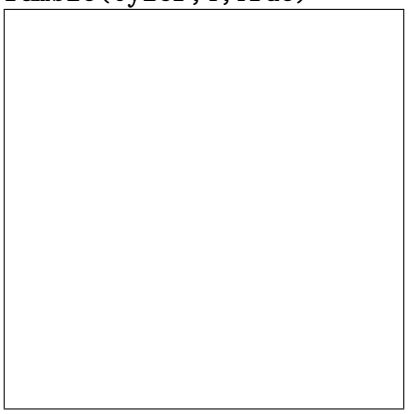


4. (a) Draw the output for the function calls:

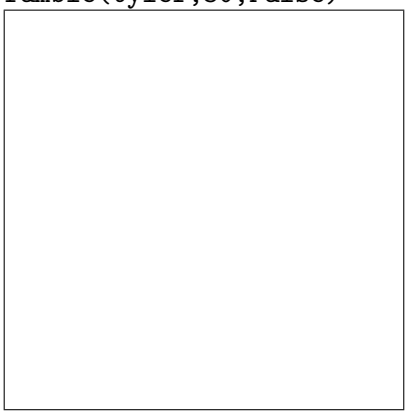
```
import turtle
tyler = turtle.Turtle()
tyler.shape('circle')

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()
```

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



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



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

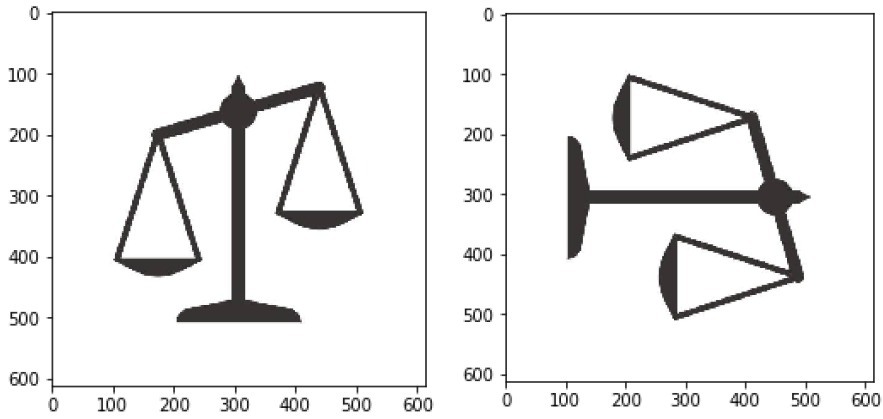
(c) If you call `ramble(tyler,4,True)`, which branches of the function are tested:

- ☐ 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:

- ☐ 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 hight and length)



Libraries:

Input:

Output:

Process (as a list of steps):

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 tl	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:

```
#P6,V3: extracts dates with highest number of troll tweets
#Import the libraries for data frames and plotting data:
```

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

```
csvFile =
```

```
#Read input data into data frame:
```

```
trolls =
```

```
#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 =
```

```
#Print the top 5 dates with most troll tweets
```

```
print(trollDates[
```

```
:
```

```
] )
```

```
#Generate a bar plot of the top 5 dates with largest number of troll tweets
```

```
trollDates.
```

```
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%.

```
#Import the packages for images and arrays:
```

```
#Ask user for image name and read into img:
```

```
#Get height and width:
```

```
#Initialize counter:
```

```
#Loop through all the pixels & update count if very light:
```

```
#Compute and print fraction:
```

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

Output:

(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.

```
☐ ADDI $sp, $sp, -6          # Set up stack

☐ ADDI $s3, $zero, 1         # Store 1 in a registrar

☐ ADDI $t0, $zero, 33        # Set $t0 at 33 (!)

☐ ADDI $s2, $zero, 5         # Use to test when you reach 5

☐ 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, -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
```

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

(a)

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

Output:

(b)

```
#include <iostream>
using namespace std;
int main()
{
    double num = 0;
    double weight = 0;
    while (weight < 100) {
        cout <<"Please enter weight\n";
        cin >> weight;
        num += weight;
    }
    cout << num << endl;
    return 0;
}
```

Input: 50,75,150
Output:

(c)

```
#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)
                cout << "O";
            else
                cout << "X";
        }
        cout << endl;
    }
    return 0;
}
```

Output:

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

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

```
//include library and namespace
```

```
//function signature
```

```
{
  //loop line
```

```
  //loop body
```

```
  //return
```

```
}
```

(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.

```
//include library and namespace
```

```
//function signature
```

```
{
  //initialize variables
```

```
  //obtain input
```

```
  //calculate users
```

```
  //output users
```

```
  //return
```

```
}
```