Row:	Seat:

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

20 December 2021

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) Fill in the code below to produce the Output on the right:

```
pioneers = "Lovelace,Ada-Fleming,Williamina-Hopper,Grace"
```

i.
```
names =
```
[ ]

```
for n in names:
```

ii.
```
    print(
```
[ ]
```
)
```

Output:

```
Lovelace
Fleming
Hopper
```

(b) Consider the following shell commands:

```
$ pwd
/usr/student
$ ls
classes.csv grades.csv  hello.py  hw60.py
```

i. What is the output for:

```
$ mkdir projects
$ mv *py projects
$ ls
```

Output:

ii. What is the output for:

```
$ cd projects
$ ls | grep hw
```

Output:

iii. What is the output for:

```
$ cd ../
$ pwd
```

Output:

2. (a) Select the color corresponding to the rgb values below:

   i. `rgb = (255, 0, 0)`
   ☐ black ☐ red ☐ white ☐ gray ☐ purple

   ii. `rgb = "#AB00AB"`
   ☐ black ☐ red ☐ white ☐ gray ☐ purple

   iii. `rgb = (0.5, 0.5, 0.5)`
   ☐ black ☐ red ☐ white ☐ gray ☐ purple

   iv. What is the 5-bit binary number equivalent of Decimal 24?

   `Decimal 24 = Binary` | | | | | |

   v. What is the Decimal number equivalent to Hexadecimal 1C?

   `Hexadecmal 1C = Decimal` | | |

(b) Given the list `symbols` below, fill in the code to produce the Output on the right:

```
symbols = ['*', '#', '+', '$', '%']

for i in range( [ ] ):
```

i.

```
    for j in range( [ ] ):
        print(symbols[j], end=" ")
```

Output:

```
* # + $ * # + $
```

ii.

```
for j in range( [ ] , [ ] , [ ] ):
    print(symbols[j], end=" ")
```

Output:

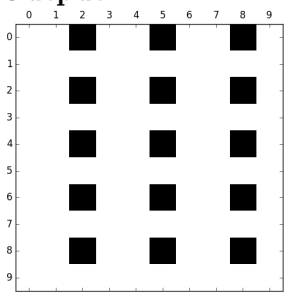
```
% + *
```

iii.

```
import numpy as np
import matplotlib.pyplot as plt
im = np.ones( (10,10,3) )

im[0::[ ] , 2::[ ] , :]  = 0
plt.imshow(im)
plt.show()
```

Output:



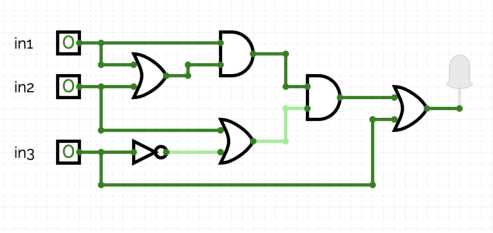
3. (a) What is the value (True/False):

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

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

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

iv.



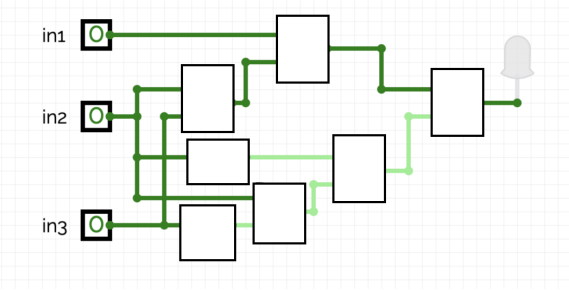

```
in1 = True
in2 = False
in3 = False
```
☐ True ☐ False

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

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

(c) Fill in the circuit with the gate-symbol or gate-name that implements the logical expression:

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



4. Consider the following functions:

```
import numpy as np

def find_all(grid, n):
  for i in range(grid.shape[0]):
    for j in range(grid.shape[1]):
      if compare(grid[i,j], n):
        print(grid[i,j])
```

```
def compare(x, num):
   return x % num == 0

def main():
   table = np.array([[1, 2, 3, 4],
                     [15, 20, 25, 30],
                     [5, 10, 50, 75]])
   find_all(table, 10)
```

(a) What are the formal parameters for `compare()`?

(b) What are the actual parameters for `find_all()`?

(c) How many calls are made to `compare()` after calling `main()`?

(d) What is the output after calling `main()`?

i. **Output:**

5. Design an algorithm that, given an image, outputs the number of pixels that are considered dark based on some user-provided threshold for darkness.

Libraries:

Input:

Output:

Design Pattern:
☐ Search ☐ Find Min ☐ Find Max ☐ Find All

Principal Mechanisms (select all that apply):
☐ Single Loop ☐ Nested Loop ☐ Conditional (if/else) statement
☐ Indexing / Slicing ☐ `split()` ☐ `groupby()`

Process (as a concise and precise LIST OF STEPS / pseudocode):
(Assume libraries have already been imported.)

6. Consider the `covid_19.csv` dataset that reports the number of observed COVID-19 cases in different countries by observation date. A snapshot given in the image below:

covid_19_data

ObservationDate	Province/State	Country/Region	Last Update	Confirmed	Deaths	Recovered
07/16/2020	Shizuoka	Japan	2020-07-17 04:34:50	100.0	1.0	83.0
07/16/2020	Sichuan	Mainland China	2020-07-17 04:34:50	599.0	3.0	590.0
07/16/2020	Sicilia	Italy	2020-07-17 04:34:50	3132.0	283.0	2695.0
07/16/2020	Sinaloa	Mexico	2020-07-17 04:34:50	10859.0	1739.0	8572.0
07/16/2020	Sindh	Pakistan	2020-07-17 04:34:50	108913.0	1888.0	70292.0
07/16/2020	Sint Maarten	Netherlands	2020-07-17 04:34:50	78.0	15.0	63.0
07/16/2020	Smolensk Oblast	Russia	2020-07-17 04:34:50	5262.0	83.0	3180.0
07/16/2020	Sonora	Mexico	2020-07-17 04:34:50	13315.0	1235.0	11423.0

Fill in the Python program below:

```
#Plots number of recovered cases in Italy by observation date
#Import the libraries for data frames and plotting data

#Prompt user for input file name:

csvFile =

#Read input data into data frame:

df =

#Groups the data by Country/Region to extract observations in Italy

italy =

#Plot the number of recovered cases over time (observation date)

italy.plot(  )

plt.show()
```

7. Write a **complete Python program** that prompts the user for the name of a .csv file and the names of latitude and longitude columns and generates an interactive .html map with markers found at each geographical location extrated from the .csv file.

```
#Import the packages for dataframes and for generating html maps
```

```
#Ask user for the name of csv file and store in variable in_file
```

```
#Ask user for the name of latitude and longitude columns
#and store in variables lat and long respectively
```

```
#Read the csv file into a dataframe and store it in variable df
```

```
#Create a map and store in variable map
```

```
#Loop through all the rows in the dataframe, create a marker with
#values found in columns lat and long, add marker to the map
```

```
#Save the map to file named map.html
```

8. (a) What does the MIPS program below print:

Output:

(b) Modify the program to print out 15 consecutive letters in decreasing order ('Z' down to 'L'). Shade in the box for each line that needs to be changed and rewrite the instruction below.

☐ `ADDI $sp, $sp, -8 # Set up stack`

☐ `ADDI $s3, $zero, 1 # Store 1 in a register`

☐ `ADDI $t0, $zero, 97 # Set $t0 at 97 (a)`

☐ `ADDI $s2, $zero, 7 # Use to test when you reach 7`

☐ `SETUP: SB $t0, 0($sp) # Next letter in $t0`

☐ `ADDI $sp, $sp, 1 # Increment the stack`

☐ `SUB $s2, $s2, $s3 # Decrement the counter by 1`

☐ `ADDI $t0, $t0, 1 # Increment the letter`

☐ `BEQ $s2, $zero, DONE # Jump to DONE if s2 == 0`

☐ `J SETUP # Else, jump back to SETUP`

☐ `DONE: ADDI $t0, $zero, 0 # Null (0) to terminate string`

☐ `SB $t0, 0($sp) # Add null to stack`

☐ `ADDI $sp, $sp, -7 # Set up stack to print`

☐ `ADDI $v0, $zero, 4 # 4 is for print string`

☐ `ADDI $a0, $sp, 0 # Set $a0 to stack pointer`

☐ `syscall # Print to the log`

9. Fill in the C++ programs below to produce the Output on the right.

(a)

```
#include <iostream>
using namespace std;
int main()
{
    int num = [          ];
    for(int i = 0; i <=30; [          ]){
       num += 5;
       cout << i << " " << num << endl;
    }
    return 0;
}
```

Output:

```
0 10
10 15
25 20
```

(b)

```
#include <iostream>
using namespace std;
int main()
{
    double num = 0;
    double tot = 0;

    while ( [          ]){
        cout <<"Please enter amount\n";
        cin >> num;
        tot += num;
    }
    cout <<"The total is " <<  tot << endl;
    return 0;
}
```

Input: 5, 3, 2, 1
Output:

```
Please enter amount
5
Please enter amount
3
Please enter amount
2
Please enter amount
1
The total is 11
```

(c)

```
#include <iostream>
using namespace std;
int main(){

    for (int i = 1; [          ] i++){

        for (int j = 0; [          ] j++ ){
            if(j % 2 == 0)
                cout << "X";
            else
                cout << "O";
        }
        cout << endl;
    }
    return 0;
}
```

Output:

```
X
XO
XOX
XOXO
```

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

```
#Python Loops
for i in range(0,101,25):
    print(i+5, i-5)
```

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

```
//main function signature
```

```
{
  //loop line
```

```
  //loop body
```

```
  //return
```

```
}
```

(b) Parsec is a unit of distance used in astronomy, equal to 3.26 light years and 30.9 trillion kilometers. One parsec corresponds to the distance at which the mean radius of the earth's orbit subtends an angle of one second of arc.
Write a **complete C++ program** that asks the user for the number of parsecs and prints the corresponding number of light years and kilometers.

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

```
//main function signature
```

```
{
  //initialize variables
```

```
  //obtain input
```

```
  //calculate conversions
```

```
  //output conversions
```

```
  //return
```

```
}
```