

Row:	SEAT:

FINAL EXAMINATION, VERSION 4
 CSci 127: Introduction to Computer Science
 Hunter College, City University of New York

May 2026

Exam Rules

- Show all your work. Your grade will be based on the work shown.
- You may have pens, pencils and one 8 1/2" x 11" reference sheet filled with notes. No other materials are allowed.
- No phones, computers, tablets, calculators, watches, smart glasses, smart pencils, earpods, or other electronic devices are allowed.
- All electronic devices must be turned off and stored in your bag. If you are not able to turn off the Bluetooth/Wifi on your device, put it in your bag at the front of the room.
- **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:									
EmplID:									
Email:									
Signature:									

1. (a) Assuming the code is run sequentially, what is the output at each step:

```
stops=["Jct Blvd", "90 St", "82 St", \
i.     "74 St", "69 St", "61 St"]
print(len(stops), "stops")
```

Output:

```
ii. print("Last:", stops[-1])
```

Output:

```
trs = ["Q72", "", "", "E,F,M,R", "", \
iii.  "Q70 SBS"]
jct_tr = trs[0]
print(jct_tr)
```

Output:

```
nums = [t.count(",") for t in trs]
iv.   for s,t,n in zip(stops,trs,nums):
       if len(t) > 0:
           print(s,":",n+1,"transfers.")
```

Output:

- (b) Consider the following shell commands:

```
$ ls -l
-rw-r--r--  1 stjohn  staff   19086 Jan 13 16:18 lab8.html
-rw-r--r--  1 stjohn  staff   21510 May  5 14:53 lab9.html
-rw-r--r--  1 stjohn  staff   93628 Apr  1 16:35 ps.html
-rw-r--r--  1 stjohn  staff    4101 Jan 21 07:42 resources.html
-rw-r--r--  1 stjohn  staff   17283 Jan  9 14:59 syl.html
```

Assuming the commands below are run sequentially, what is the output after each has run:

i. \$ ls

Output:

ii. \$ ls -l | grep "Apr" | wc -l

Output:

iii. \$ echo "Labs:"
\$ ls l*.html | wc -l

Output:

iv. \$ mkdir labs
\$ mv l*.html labs
\$ ls

Output:

2. (a) Check **all that apply**:

i. What color is `tess` after this command? `tess.color("#0000DA")`

- black white red blue None of these

ii. Select all the binary numbers **larger than 8**:

- 0010 1101 0111 0110 1100

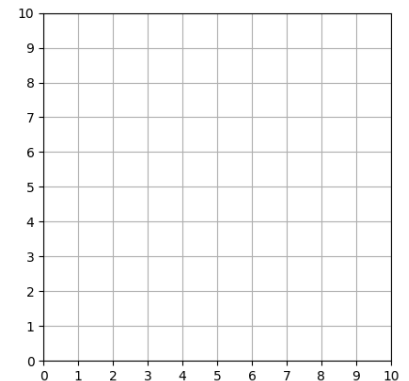
iii. Select the **odd hexadecimal numbers**:

- C C0 C7 89 9F

(b) Show the output of this program by shading in the grid:

```
import matplotlib.pyplot as plt
import numpy as np
logo = np.ones( (10,10,3) )
logo[0:2,:,:] = 0
logo[:,0:2,:] = 0
logo[-2,:,:] = 0
plt.imshow(logo, extent= [0,10,0,10])
plt.show()
```

Output:



(c) For each error below, give the line number and the code that would fix the error.

```
1 import pandas as pd
2 irt_data = {'Lines' : [1,2,3,4,5,6,7], 'Colors' : ['R','R','R','G','G','G','P']}
3 print(irt_data items())
4 df = pd.DataFrame(irt_dat)
5 print( df['Colors'] )
```

i. `print(irt_data items())`
`~~~~~`
 SyntaxError: invalid syntax.

Line Number: **Correct code:**

ii. `df = pd.DataFrame(irt_dat)`
`~~~~~`
 NameError: name 'irt_dat' is not defined.

Line Number: **Code that fixes the error:**

iii. `print(df['Colors'])`
`~`
 SyntaxError: closing parenthesis ']' does not match opening parenthesis '('

Line Number: **Correct code:**

3. (a) What will make the following statement true: **Check all that apply.**

$in1 \text{ or } not \text{ } in2$

- Setting $in1 = \text{False}$ and $in2 = \text{False}$.
- Setting $in1 = \text{False}$ and $in2 = \text{True}$.
- All values for $in1$ and $in2$ make the statement true.
- Setting $in1 = \text{True}$ and $in2 = \text{False}$.
- Setting $in1 = \text{True}$ and $in2 = \text{True}$.
- No values for $in1$ and $in2$ make the statement true.

(b) What is the value of out?

$in1 = \text{False}$
 $in2 = \text{True}$
 $in3 = \text{True}$
 $out = (in1 \text{ and } in2) \text{ or } not \text{ } in3$

Choose one:

- False
- True
- Not defined

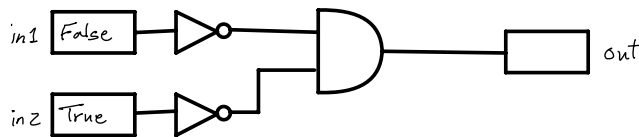
(c) Fill in the values to yield the output:

$in1 =$
 $in2 =$

$out =$

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

(d) What is the output of this circuit?

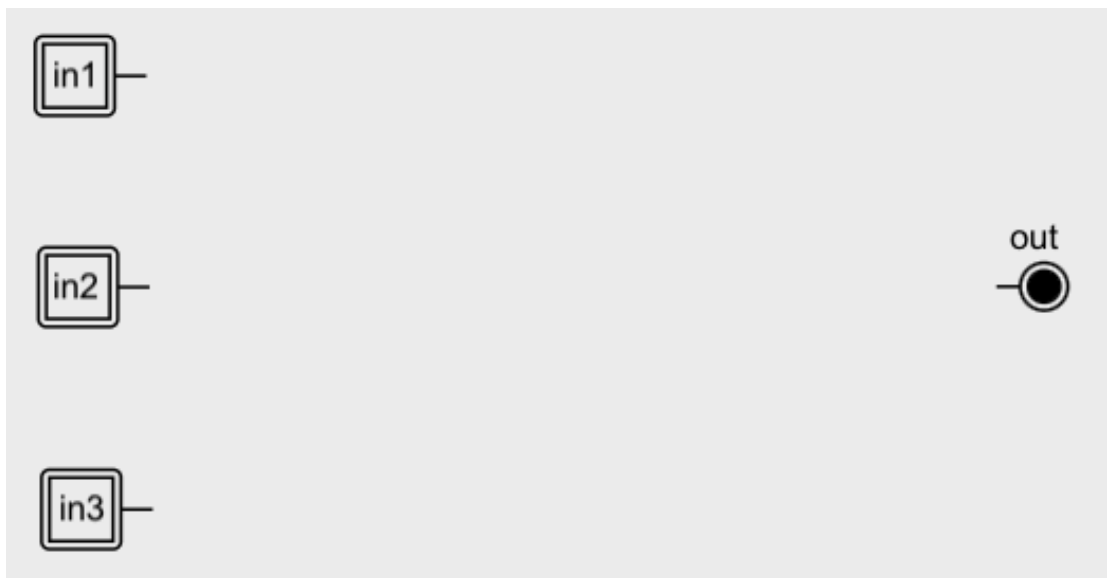


Choose one:

- False
- True
- Not defined

(e) Design a circuit that **exactly implements** the logical expression:

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



4. (a) Using the turtle and function below, fill in the parameters that yield the output:

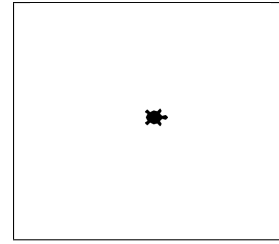
```

1 import turtle
2 thea = turtle.Turtle()
3 thea.shape("turtle")
4
5 def ramble(tur, length):
6     if length >= 20:
7         for i in range(4):
8             tur.forward(length)
9             tur.right(360/4)
10            ramble(tur, length/2)
11     elif length < 0:
12         print("Error: neg. length!")

```

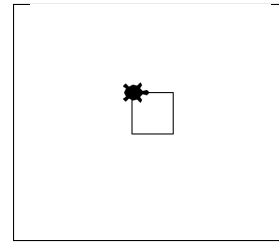
i. `ramble(` `)`

Output:



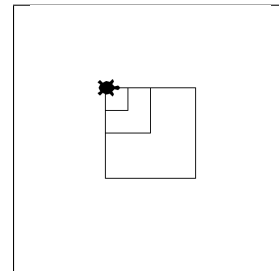
ii. `ramble(` `)`

Output:



iii. `ramble(` `)`

Output:



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

(c) For which values of `num`, will `ramble(thea,num)` print Error (check all that apply):

`num = -10`

`num = 100`

`num = 0`

None of the above.

`num = 10`

(d) For which values of `num`, will `ramble(thea,num)` **NOT** execute Line 7 (check all that apply):

`num = -10`

`num = 100`

`num = 0`

None of the above.

`num = 10`

5. Write a function `max_magnitude()` that takes a list of integers and returns the largest absolute value of a number in that list. For example:

```
numbers = [1, -4, 55, 6, 2, 4, -90, 4, 2, 3]
print(find_highest(numbers))
```

would print 90 since it has the largest absolute value (e.g. ignore signs) in the inputted list.

Libraries:	
Input:	
Output:	

Design Pattern:

- Accumulator Max/Min Finding Duplicates Searching

Principal Mechanisms (select all that apply):

- Single Loop Nested Loop Conditional (if/else) Recursion
 Indexing/slicing Dictionary List Comprehension Regular Expressions

Process (as a concise and precise LIST OF STEPS / pseudocode):

(Assume libraries have already been imported.)

6. Fill in the Python code below that creates an interactive map using Plotly Express.

#Libraries used:

#Read in names, latitudes, and longitudes into separate variables:

name_str =

lat_str =

lon_str =

#Split up the inputted strings into lists:

names =

lats =

lons =

#Set up a dictionary of the lists (used to make df):

#Make a DataFrame of the dictionary:

#Use column names of df for keyword args:

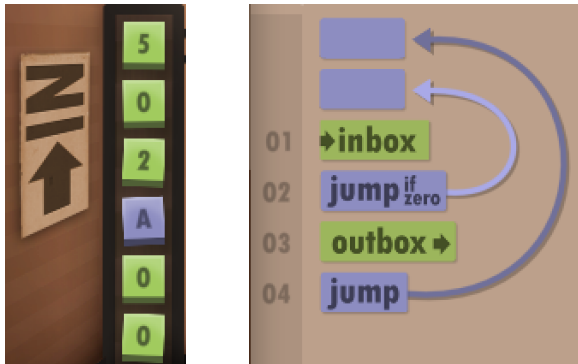
#Ask for output file name:

#Write DataFrame to specified file:

7. Write a **complete Python program** that

- asks the user for the name of a png file and
- prints the number of pixels that are bright blue (the fraction of blue is above 0.75 and the fraction of green, and the fraction of red are below 0.25).

8. (a) What does the Human Resource Machine (HRM) code output with the following input:



Output:

Note: if the input is a letter, it is not equal to zero.

(b) Consider the following MIPS program:

```

ADDI $s0, $zero, 20
ADDI $s1, $zero, 10
ADD $s2, $s0, $s1
SUB $s3, $s2, $s1
    
```

After the program runs, what is the value stored in:

\$s0 register	\$s1 register	\$s2 register	\$s3 register

(c) Consider the MIPS code:

```

1  ADDI $sp, $sp, -7
2  ADDI $t0, $zero, 65
3  ADDI $t1, $zero, 97
4  ADDI $s2, $zero, 3
5  SETUP: SB $t0, 0($sp)
6  ADDI $sp, $sp, 1
7  SB $t1, 0($sp)
8  ADDI $sp, $sp, 1
9  ADDI $s2, $s2, -1
10 BEQ $s2, $zero, DONE
11 J SETUP
12 DONE: ADDI $t0, $zero, 0
13 SB $t0, 0($sp)
14 ADDI $sp, $sp, -6
15 ADDI $v0, $zero, 4
16 ADDI $a0, $sp, 0
17 syscall
    
```

i) How many characters are printed?	
ii) What is the first character printed?	
iii) What is the message printed?	
iv) List what you need to change to print the message twice:	

9. (a) Fill in the missing code to yield the output:

```
//Lyrics by Lopez & Lopez
#include <iostream>
using namespace std;
int main(){
    cout << "It's time to see what ";
    
    cout << "limits and break through";
    cout << endl;
    return(0);
}
```

Output:

It's time to see what I can do
To test the limits and break through

- (b) What is the output:

```
//More Elsa
#include <iostream>
using namespace std;
int main()
{
    int count = 2;
    while (count > 0) {
        cout <<"Let it go, ";
        count--;
    }
    cout << "\nI am one with ";
    cout << "the wind and sky\n";
    return(0);
}
```

Output:

- (c) What is the output:

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

Output:

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

Python program:

```
temp=int(input('Enter temperature today:'))
if temp <= 32:
    print('Freezing!')
else:
    celsius = (temp-32)*5/9
    print(celsius, 'degrees')
```

C++ program:

- (b) Write a complete C++ program that prints the change in population of the the United States: $p = p + B*p - D*p$, where p is the population, B is the birth rate of 12.4 births for every 1000 people (12.4/1000) each year, and D is the death rate of 8.4 for every 1000 people (8.4/1000). In 2017, the population of United States was 325.7 million. Your program should ask the user for the number of years and print expected population over those years starting from 2017. Each line should have: the year and the population (in millions). A sample run:

```
Please enter the number of years: 5
Year 2017  325.70
Year 2018  327.00
Year 2019  328.31
Year 2020  329.62
Year 2021  330.94
```

SCRATCH PAPER

SCRATCH PAPER

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]