

Answers, inline, preceded by red boxes. See exam for full questions and formatting.

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

May 19, 2026

1. (a) What will the following Python code print:
- ```
s = "the_City_University_of_New_York"
num = s.count("_")
print('Number is', num)
words = s.split("_")
print('words[1] is', words[1].upper())
actual=[w for w in words if ord(w[0])<91]
print(actual)
short = ""
for word in actual:
 print(word[0], word)
 short = short + word[0]
print("Short version: ", short)
print("Of", words[-2:])
```
- Number is 5  
words[1] is CITY  
['City', 'University', 'New', 'York']  
C City  
U University  
N New  
Y York  
Short version: CUNY  
Of ['New', 'York']

- (b) Consider the following shell commands:

```
$ ls
p1.py p2.py p50.cpp photos world_map.html
$ pwd
/Users/csguest
```

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

- i. \$ mv world\_map.html photos  
\$ ls  
p1.py p2.py p50.cpp photos

```

ii. $ touch p3.py
 $ ls p* | wc -l
 5
 $ mkdir hw
iii. $ mv p*.py hw
 $ ls
 hw p50.cpp photos

 $ cd hw
iv. $ echo "hw:"
 $ pwd
 hw:
 /Users/csguest/hw

```

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

- i. What color is `tess` after this command? `tess.color("#00CD00")`  
 black       white       red       blue       None of these
- ii. Select all the binary numbers **smaller than 8 decimal**:  
 1001       1101       0111       1010       0110
- iii. Select the **even** hexadecimal numbers:  
 A       B       90       99       FF

(b) Fill in the missing code to yield the output:

```

import matplotlib.pyplot as plt
import numpy as np
#Set background color to white:

logo = ((10,10,3))
#Draw letter in black:
logo[:,0:2,:] = 0

logo[:,,:] = 0

logo [,:,:] = 0
plt.imshow(logo)
plt.show()

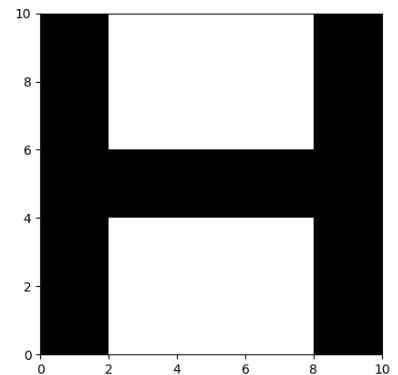
```

```

import matplotlib.pyplot as plt
import numpy as np
#Set background color to white:
logo = np.ones((10,10,3))
#Draw letter in black:
logo[:,0:2,:] = 0
logo[:,-2:,:] = 0
logo[4:6,:,:] = 0
plt.imshow(logo)
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 fn = 'data.csv"
3 df = read_csv(fn)
4 for c in df.columns:
5 print(c.upper())
```

i. `fn = 'data.csv"`

^

SyntaxError: unterminated string literal

Line Number: Correct code:

2

`fn = 'data.csv'`

ii. `df = read_csv(fn)`

^^^^^^^^

NameError: name 'read\_csv' is not defined

Line Number: Correct code:

3

`df = pd.read_csv(fn)`

iii. `print(c.upper())`

^

SyntaxError: '(' was never closed

Line Number: Correct code:

5

`print(c.upper())`

3. (a) What will make the following statement true:

`not in1 and in1`

- Only setting `in1 = False`.  Neither value for `in1` makes the statement true.
- Both values for `in1` make the statement true.
- Only setting `in1 = True`.

(b) What is the value of `out`?

```
in1 = False
in2 = True
out = not (in1 and not in2)
out = True
```

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

`in1 =`

|      |
|------|
| True |
| True |

`in2 =`

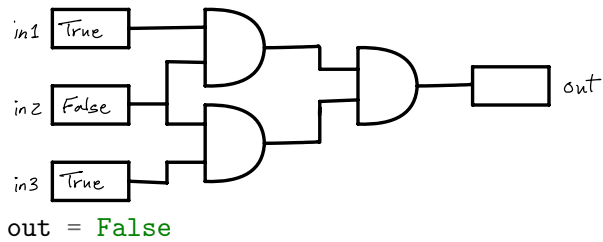
|      |
|------|
| True |
|------|

`out =`

|      |
|------|
| True |
|------|

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

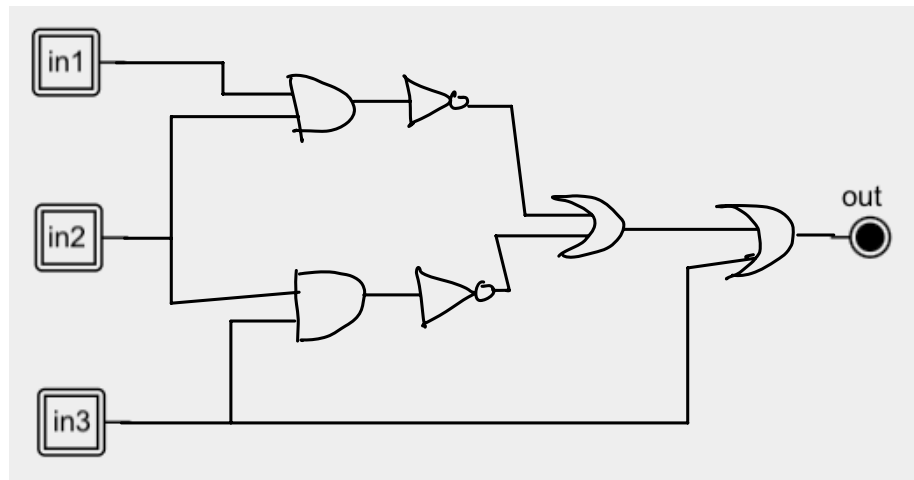
(d) What is the output of this circuit?



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

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

$\sqrt{0}$ :  $(\text{not}(\text{in1 and in2})) \text{ or } (\text{not}(\text{in2 and not in3})) \text{ or in3}$



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

```

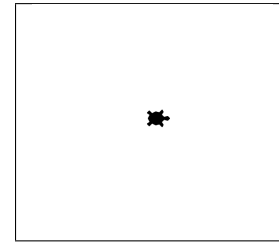
1 import turtle
2 tess = turtle.Turtle()
3 tess.pensize(5)
4 tess.shape("turtle")
5
6 def drawT(t, commands):
7 if len(commands) != 0:
8 ch = commands[0]
9 if ch == 'F':
10 t.forward(50)
11 elif ch == 'L':
12 t.left(90)
13 elif ch == 'R':
14 t.right(90)
15 elif ch == '^':
16 t.penup()
17 elif ch == 'v':
18 t.pendown()
19 elif ch == 'S':
20 t.stamp()
21 else:
22 print("Error:", ch)
23 drawT(t,commands[1:])

```

i. drawT( )

tess, "S"

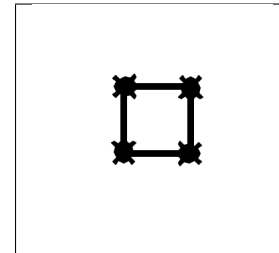
Output:



ii. drawT( )

tess, "SFRSFRSFRSFR"

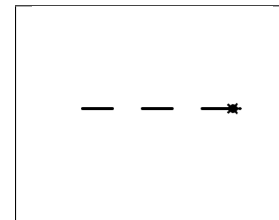
Output:



iii. drawT( )

tess, "F^FvF^FvF"

Output:



(b) What are the formal parameters for the function drawT():  
t, commands

(c) For which values of sss, will drawT(tess,sss) print Error (check all that apply):

- |                                               |                                                 |
|-----------------------------------------------|-------------------------------------------------|
| <input type="checkbox"/> sss = "L^FRvF^R"     | <input checked="" type="checkbox"/> sss = "BFL" |
| <input checked="" type="checkbox"/> sss = "s" | <input type="checkbox"/> None of the above.     |
| <input type="checkbox"/> sss = "FFFF"         |                                                 |

(d) For which values of sss, will drawT(tess,sss) **NOT** execute Line 23 (check all that apply):

- |                                              |                                             |
|----------------------------------------------|---------------------------------------------|
| <input checked="" type="checkbox"/> sss = "" | <input type="checkbox"/> sss = "^SFSFSv"    |
| <input type="checkbox"/> sss = "sS"          | <input type="checkbox"/> None of the above. |
| <input type="checkbox"/> sss = "FFFF"        |                                             |

5. Design an algorithm that asks the user for a list of locations (given as latitude, longitude, and

name) they have visited. Your program should keep track of how many times each location occurs and create only one marker for each location, with the hover text containing the number of visits.

For example, if the inputted list is:

```
[(40.7678,-73.9645,'Hunter College'),(40.6892,-74.0445,'Statue of Liberty'),\
(40.7678,-73.9645,'Hunter College')]
```

your map would display 2 markers: one for Hunter College with a hover text of 2, and one for Statue of Liberty with a hover text of 1.

|                   |                                                                    |
|-------------------|--------------------------------------------------------------------|
| <b>Libraries:</b> | pandas, plotly.express                                             |
| <b>Input:</b>     | a list (containing tuples of the latitudes, longitudes, and names) |
| <b>Output:</b>    | an HTML file containing a map                                      |

### Design Pattern:

Accumulator    Max/Min    Finding Duplicates    Searching

### Principal Mechanisms (select all that apply):

Loop    Conditional (if/else)    Recursion    Indexing/slicing  
 `input()`    Dictionary    List Comprehension    Regular Expressions

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

(Assume libraries have already been imported.)

- (a) Input the list, `locations` from the user.
  - (b) Set up an empty dictionary, `counts`.
  - (c) For `loc` in `locations`:
  - (d)   Check if the `loc` is in the dictionary.
  - (e)   If it is, increment the count ( `counts[loc] += 1` ).
  - (f)   If it isn't, add the location with value 1 to the dictionary ( `counts[loc] = 1` ).
  - (g) Create a DataFrame, using the unique locations and counts. This can be done multiple ways: one way:
    - i. Set up empty lists for the `lats`, `lons`, `names`, and `counts`.
    - ii. For each item in `locations`, append its values to these lists.
    - iii. Make a new dictionary:
 

```
{'Latitude' : lats, 'Longitude': lons, 'Name': names, 'Count' : counts}
```
    - iv. Create a DataFrame, `df`, from the dictionary.
  - (h) Use the DataFrame, `df`, as the data for Plotly Expresses, `px.scatter_map()` to create the map.
  - (i) Display/save the map.
6. Fill in the following functions that are part of a program that draws with turtles:
- `getData()`: gets the color and shape of a turtle and the number of sides of a polygon
  - `getTurtle()`: returns a turtle with color and shape
  - `drawPolygon()`: draws a polygon with `n` sides using turtle `t`

```

import turtle
def getData():
 """
 Asks the user for the color and shape of a turtle
 and the number of sides of a polygon.
 Returns the color and shape as strings and the sides as integer.
 """
 col = input('Enter the color of the turtle ')
 sha = input('Enter the shape of the turtle ')
 sides = int(input('Enter the sides of a polygon '))
 return(col, sha, sides)

def getTurtle(tcol, tsha):
 """
 Returns a turtle with color, tcol, and shape, tsha
 """
 t = turtle.Turtle()
 t.color(tcol)
 t.shape(tsha)
 return(t)

def drawPolygon(t, n):
 """
 Draws a polygon with n sides using turtle t
 """
 for i in range(n):
 t.forward(100)
 t.left(360/n)

```

7. Write a **complete program**, based on the Queens Library dataset. Your program should:

- Ask for the name of the input CSV file and the name of the output HTML file.
- Use Plotly Express to create a map with the libraries open on Sunday. When the user hovers over the marker, the Sunday hours (e.g. 'Su' column) should appear in the pop-up.
- Save the map to output file specified by the user.

First rows of Queens Library dataset:

| name              | Address                   | City         | Mn               | Tu               | We               | Th               | Fr               | Sa              | Su              |
|-------------------|---------------------------|--------------|------------------|------------------|------------------|------------------|------------------|-----------------|-----------------|
| Rosedale          | 144-20 243 Street         | Rosedale     | 12:00 PM - 8:00  | 1:00 PM - 6:00 F | 10:00 AM - 6:00  | 12:00 PM - 8:00  | 10:00 AM - 6:00  | 10:00 AM - 5:00 | Closed - Closed |
| Bayside           | 214-20 Northern Boulevard | Bayside      | 9:00 AM - 8:00 F | 1:00 PM - 6:00 F | 10:00 AM - 6:00  | 12:00 PM - 8:00  | 10:00 AM - 6:00  | 10:00 AM - 5:00 | Closed - Closed |
| Central Library-  | 89-11 Merrick Boulevard   | Jamaica      | 9:00 AM - 9:00 F | 9:00 AM - 7:00 F | 9:00 AM - 7:00 F | 9:00 AM - 7:00 F | 9:00 AM - 7:00 F | 10:00 AM - 5:00 | 12:00 PM - 5:00 |
| Hollis            | 202-05 Hillside Avenue    | Hollis       | 12:00 PM - 8:00  | 1:00 PM - 6:00 F | 10:00 AM - 6:00  | 12:00 PM - 8:00  | 10:00 AM - 6:00  | 10:00 AM - 5:00 | Closed - Closed |
| Queens Library f  | 2002 Cornaga Avenue       | Far Rockaway | 2:30 PM - 6:00 F | 2:30 PM - 6:00 F | 2:30 PM - 6:00 F | 2:30 PM - 6:00 F | 2:30 PM - 6:00 F | Closed - Closed | Closed - Closed |
| Douglaston/Little | 249-01 Northern Boulevard | Little Neck  | 12:00 PM - 8:00  | 1:00 PM - 6:00 F | 10:00 AM - 6:00  | 12:00 PM - 8:00  | 10:00 AM - 6:00  | 10:00 AM - 5:00 | Closed - Closed |
| Queensboro Hill   | 60-05 Main Street         | Flushing     | 12:00 PM - 8:00  | 1:00 PM - 6:00 F | 10:00 AM - 6:00  | 12:00 PM - 8:00  | 10:00 AM - 6:00  | 10:00 AM - 5:00 | Closed - Closed |
| Howard Beach      | 92-06 156 Avenue          | Howard Beach | 12:00 PM - 8:00  | 1:00 PM - 6:00 F | 10:00 AM - 6:00  | 12:00 PM - 8:00  | 10:00 AM - 6:00  | 10:00 AM - 5:00 | Closed - Closed |

```

import plotly.express as px
import pandas as pd

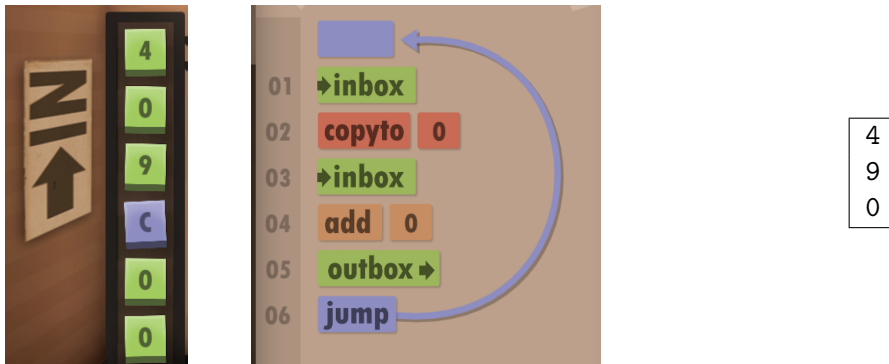
```

```

in_file = input('Enter input file: ')
q_lib_df = pd.read_csv(in_file)
q_lib_df = q_lib_df[q_lib_df["Su"] != "Closed - Closed"]
fig = px.scatter_map(q_lib_df,
 lat="Latitude",
 lon="Longitude",
 hover_name="Su",
 title="Queens Library Branches",
 zoom=10)
out_file = input('Enter output file name: ')
fig.write_html(out_file)

```

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



*Note: if the input is a letter rather than a number, it counts as zero.*

- (b) Consider the following MIPS program:

```

ADDI $s0, $zero, 2
ADDI $s1, $zero, 4
SUB $s2, $s1, $s0
ADD $s3, $s1, $s0

```

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

| \$s0 register | \$s1 register | \$s2 register | \$s3 register |
|---------------|---------------|---------------|---------------|
| 2             | 4             | 2             | 6             |

- (c) Consider the MIPS code:

```

1 ADDI $sp, $sp, -6
2 ADDI $t0, $zero, 72
3 ADDI $s2, $zero, 67
4 SETUP: SB $t0, 0($sp)
5 ADDI $sp, $sp, 1
6 ADDI $t0, $t0, -1
7 BEQ $t0, $s2, DONE
8 J SETUP

```

EmplID:

|                                          |                                   |
|------------------------------------------|-----------------------------------|
| ii) What is the first character printed? | 11                                |
| iii) What is the message printed?        | HGFED<br>CSci 127 Mock Final, S26 |

```
9 DONE: ADDI $t0, $zero, 0
10 SB $t0, 0($sp)
11 ADDI $sp, $sp, -5
12 ADDI $v0, $zero, 4
13 ADDI $a0, $sp, 0
14 syscall
```

iv) List what you need to change to print the string "HIJKL":

line 3: Change the stop value to (ASCII 'M') (e.g. ADDI \$2,zero, 77

\* line 3: Change \$s2 to hold 76 ('C').

\* line 6: Increment, instead of decrementing \$t0 (e.g. ADDI \$t0, \$t0, 1)

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

```
#include <iostream>
using namespace std;
int main()
{
 cout << "We shape out tools\nand ";

 cout << "\t--John Culkin, 1967" << endl;
 return 0;
}

cout << "thereafter they shape us." << endl;
```

**Output:**

```
We shape out tools
and thereafter they shape us.
-John Culkin, 1967
```

(b) What is the output:

```
#include <iostream>
using namespace std;
int main()
{
 for (int i = 5; i >= 0; i--){
 if (i < 3)
 cout << "step ";
 cout << i << endl;
 }
}

5
4
3
step 2
step 1
step 0
```

(c) What is the output:

```

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

```

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

C++ program:

```

#include <iostream>
using namespace std;
int main()
{
 int num = -1;

 while ((num < 0) || (num % 10 != 0)){
 {
 cout << "Enter again: ";
 cin >> num;
 }
 cout << "Your number: " << num << endl;
 return 0;
 }
}

```

Python program:

```

num = -1
while (num < 0) or (num % 10 != 0):
 num = int(input("Enter again: "))
print("Your number:", num)

```

- (b) Write a **complete** C++ program that asks the user for the starting population and prints out the yearly population until it reaches 1000. Each year the population doubles in size.

A sample run:

```

Please enter the starting population: 50
Year 0 50
Year 1 100
Year 2 200
Year 3 400
Year 4 800

```

```
#include <iostream>
using namespace std;

int main()
{
 int year = 0;
 int pop;
 cout<< "Enter starting population: ";
 cin >> pop;

 while (pop < 1000)
 {
 cout<< "Year " << year << "\t" << pop << "\n";
 pop = pop*2;
 year++;
 }
}
```