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

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

May 2026

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

i.
```
stops = ["GCT", "51 St", "59 St",\
         "68 St", "77 St", "86 St"]
print(len(stops), "stops")
```

Answer:
```
6 stops
```

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

Answer:
```
Last: 86 St
```

iii.
```
trs = ["4,5,7,S","E,F",\
       "4,5,M,N,Q,R","", "", "4,5"]
gct_tr = trs[0]
print(gct_tr)
```

Answer:
```
4,5,7,S
```

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

Answer:
```
GCT: 4 transfers.
51 St: 2 transfers.
59 St: 6 transfers.
86 St: 2 transfers.
```

(b) Consider the following shell commands:

```
$ ls -l
-rw-r--r--    1 stjohn  staff    9901 Jan 19  2025 f24.html
-rw-r--r--    1 stjohn  staff   39765 Dec  9 12:14 f25.html
-rw-r--r--    1 stjohn  staff     972 Jan 28 06:46 index.md
-rw-r--r--    1 stjohn  staff   38332 May 20  2025 s25.html
-rw-r--r--    1 stjohn  staff   40293 Apr  1 16:40 s26.html
```

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

i. `$ ls`

Answer:

ii. `$ ls | grep "html" | wc -l`

Answer:

iii.
```
$ echo "Spring:"
$ ls s*.html | wc -l
```

Answer:

```
Spring:
2
```

iv.
```
$ mkdir fall
$ mv f*.html fall
$ ls
```

Answer:

```
fall    index.md    s25.html    s26.html
```

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

Answer:

i. What color is the turtle `tom` after this command? `tom.color("#A0A0A0")`
☐ black ☐ white ☐ red ☐ blue ✓ None of these

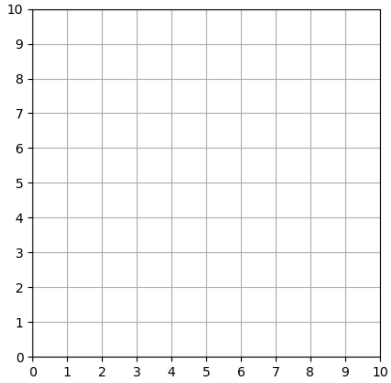
ii. Select all the binary numbers **larger than 8 decimal**:
☐ 0011 ✓ 1101 ☐ 0111 ☐ 0110 ✓ 1110

iii. Select the hexadecimal numbers **smaller than 160 decimal**:
✓ A ☐ AA ☐ B1 ✓ 99 ✓ 9F

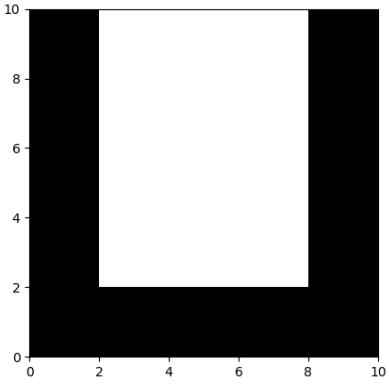
(b) Show the output of this program by shading in the grid:

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

Output:



Answer:



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

```
1 import panda as pd
2 dict = {'Boros' : ['B','Bx','M','Q','SI'] 'Pop' : [2.7,1.4,1.6,2.3,0.5]}
3 print(dict.keys()
4 df = pd DataFrame(dict)
5 print( df[['Boros']] )
```

i.
```
import panda as pd
^^^^^^^^^^^^^^^^^^
ModuleNotFoundError: No module named 'panda'
```

Answer: **Line Number:** 1 **Correct code:** `import pandas as pd`

ii.
```
print(dict.keys()
     ^
SyntaxError: '(' was never closed
```

Answer: **Line Number:** 3 **Correct code:** `print(dict.keys())`

iii.
```
df = pd DataFrame(dict)
        ^^^^^^^^^
SyntaxError: invalid syntax
```

Answer: **Line Number:** 4 **Correct code:** `df = pd.DataFrame(dict)`

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

```
in1 and not in2
```

Answer:

- ☐ Setting `in1 = False` and `in2 = False`.
- ☐ Setting `in1 = False` and `in2 = True`.
- ☐ All values for `in1` and `in2` make the statement true.
- ✓ Setting `in1 = True` and `in2 = False`.
- ☐ Setting `in1 = True` and `in2 = True`.
- ☐ No values for `in1` and `in2` make the statement true.

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

```
in1 = True
in2 = True
in3 = False
out = (in1 and in2) and not in3
```

Answer:

```
out = True
```

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

`in1 =` _____

`in2 =` _____

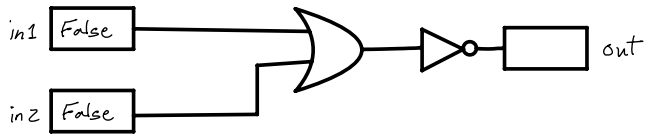
`out =` True

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

Answer:

```
in1 = False
in2 = True
```

(d) What is the output of this circuit?



Answer:

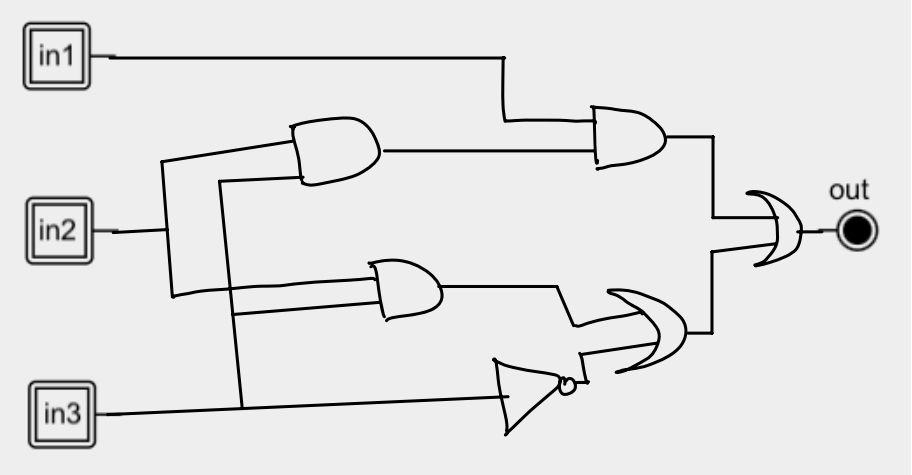
```
out = True
```

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

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

Answer:

V3: (in1 and (in2 and in3)) or ((not in2 and in3) or not in3)



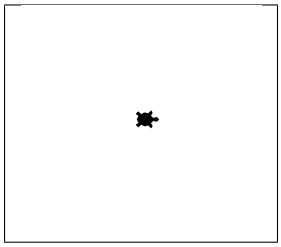
4. (a) Using the turtle and function below, fill in the parameters that yield the output:

```
1  import turtle
2  tim = turtle.Turtle()
3  tim.shape("turtle")
4
5  def ramble(t, side):
6      if side >= 20:
7          for i in range(3):
8              t.forward(side)
9              t.right(360/3)
10         ramble(t, side/2)
11     else:
12         t.stamp()
```

i. `ramble( )`

Answer: `tim, 0`

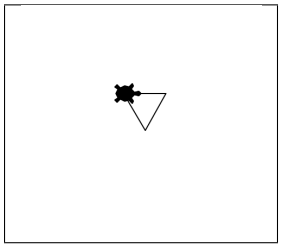
Output:



ii. `ramble( )`

Answer: `tim, 39` or any value between 20 and 39 will draw this.

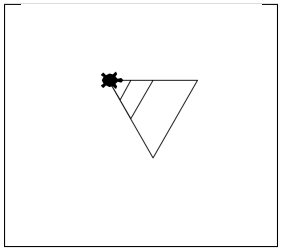
Output:



iii. `ramble( )`

Answer: `tim, 100` or any value between 80 and 159 will draw this.

Output:



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

Answer: t, side

(c) For which values of `side`, will `ramble(tim,side)` stamp a turtle (check all that apply):
Answer:

- ✓ `side = 0`
- ✓ `side = 10`
- ✓ `side = 100`
- ✓ `side = 1000`
- ☐ None of the above.

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

- ✓ `side = 0`
- ✓ `side = 10`
- ☐ `side = 100`
- ☐ `side = 1000`
- ☐ None of the above.

5. Write a function `search_for()` that takes a list of names and a target returns `True` if the name is in the list and `False` otherwise. For example:

```
names = ['Anany','Anna','Emily','Gordon','Isabelle','Jayden','Rica']
print(search_for(names, 'Anna'))
```

would print `True` since `'Anna'` in the list names.

Answer:

Libraries:	No additional– just core Python
Input:	a list of strings and a string to search for
Output:	Boolean (True/False): the target string is found

Design Pattern:

Answer:

☐ Accumulator ☐ Max/Min ☐ Finding Duplicates ✓ Searching

Principal Mechanisms (select all that apply):

Answer:

✓ 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.)

Answer:

(a) Set up variable `found = False`.
(b) For each name in the list:
(c) Check if the name matches the target
(d) If it does, set `found = True`
(e) Return `found`.

6. Fill in the missing code below to average regions of an image. For example, if you inputted our favorite image, you would see (left to right):

Answer:

```
#Fill in libraries needed for storing and displaying images:
import numpy as np
import matplotlib.pyplot as plt

def average(region):
    """
    Returns average of red values, of green values, and blue values
    across the inputted region.
    """
    red = np.mean(region[:, :, 0])
    green = np.mean(region[:, :, 1])
    blue = np.mean(region[:, :, 2])
    return (red, green, blue)
def setRegion(region, r, g, b):
    """
    Takes a region of an image and red, green, and blue values, r, g, b.
    Sets the region so that all points have
    red values of r, green values of g, and blue values of b.
    """
    region[:, :, 0] = r
    region[:, :, 1] = g
    region[:, :, 2] = b

def quarter(img2, levels):
     """
     Breaks image into regions, and calls average() and setRegion()
     to average and set colors for the regions.
     """
     hReg = img2.shape[0]//2**levels
     wReg = img2.shape[1]//2**levels
     for i in range(2**levels):
          for j in range(2**levels):
               r,g,b = average(img2[i*hReg:(i+1)*hReg,j*wReg:(j+1)*wReg])
               setRegion(img2[i*hReg:(i+1)*hReg,j*wReg:(j+1)*wReg],r,g,b)
```

7. Write a **complete Python program** that makes a DataFrame to store addresses and saves the DataFrame in a CSV file. Your program should ask the user for:

   - A list of last names,
   - A list of first names,
   - A list of emails, and
   - The name for the output (CSV) file.

   For example, a sample run of your program:

   ```
   Enter last names: Hunter Raab Kirschner Cantor
   Enter first names: Thomas Jennifer Anne Nancy
   ```

```
Enter emails: th1870@hunter jr2001@hunter ak2023@hunter nc2024@hunter
Enter file name:  addr.csv
```

would create a DataFrame:

```
        Last     First         emails
0     Hunter    Thomas  th1870@hunter
1       Raab  Jennifer  jr2001@hunter
2  Kirschner      Anne  ak2023@hunter
3     Cantor     Nancy  nc2024@hunter
```

and save the results to `addr.csv`.

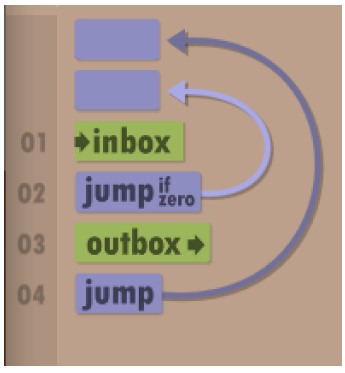
Answer:

```
import pandas as pd

last_names = input("Enter last: ").split()
first_names = input("Enter first: ").split()
emails = input("Enter email: ").split()
data = {}
data["Last"] = last_names
data["First"] = first_names
data["emails"] = emails
df = pd.DataFrame(data)

fname = input("Enter file name: ")
df.to_csv(fname)
```

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



Answer:

4
9
C

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

(b) Consider the following MIPS program:

```
ADDI $s0, $zero, 1
ADDI $s1, $zero, 2
SUB $s2, $s0, $s1
ADD $s3, $s2, $s2
```

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

$s0 register	$s1 register	$s2 register	$s3 register
Answer: 1	**Answer:** 2	**Answer:** -1	**Answer:** -2

(c) Consider the MIPS code:

```
1  ADDI $sp, $sp, -9
2  ADDI $t0, $zero, 58
3  ADDI $t1, $zero, 41
4  ADDI $s2, $zero, 4
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, -8
15 ADDI $v0, $zero, 4
16 ADDI $a0, $sp, 0
17 syscall
```

Answer:

i) How many characters are printed?	8
ii) What is the first character printed?	:
iii) What is the message printed?	:):):):)
List what you need to change to print half of the message:	* line 1: Set $sp to -5. * line 4: Set the counter $s2 to 2. * line 14: Set $sp to -4.

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 funny how some ";
    
    cout << "everything seem small" << endl;
    return(0);
}
```

Output:

It's funny how some distance
Makes everything seem small

Answer:

```
cout << "distance\nMakes ";
```

(b) What is the output:

```cpp
//More Elsa
#include <iostream>
using namespace std;
int main()
{
    int count = 2;
    while (count > 0) {
        cout <<"Let it go, ";
        count--;
    }
    cout << "\nCan't hold it ";
    cout << "back anymore\n";
    return(0);
}
```

Answer:

```
Let it go, Let it go,
Can't hold it back anymore
```

(c) What is the output:

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

Answer:

```
12345!
22222!
12345!
44444!
12345!
```

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

C++ program:

```cpp
#include <iostream>
using namespace std;
int main() {
    int times;
    cout << "Enter repetition time: ";
    cin >> times;
    for (int i = 0; i < times; i++) {
        cout<<"Practice makes perfect."<< endl;
    }
    return 0;
}
```

Python program:

Answer:

```python
times = int(input("Enter repetition time: "))
for i in range(times):
    print("Practice makes perfect.")
```

(b) Write a C++ program that asks the user for the number of credits completed and prints out number for each subsequent semester until it reaches 120. Each semester, the credits completed increases by 15.

A sample run:

```
Enter starting number of credits: 37
Semester 0:     37
Semester 1:     52
Semester 2:     67
Semester 3:     82
Semester 4:     97
Semester 5:     112
```

Answer:

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

int main()
{
    int semester = 0;
    int credits;
    cout<< "Enter starting number of credits: ";
    cin >> credits;

    while (credits < 120)
    {
        cout<< "Semester " << semester <<":\t" << credits << "\n";
        credits += 15;
        semester++;
    }
    return 0;
}
```