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

Final Examination, Version 2
CSci 127: Introduction to Computer Science
Hunter College, City University of New York

Fall 2025

1. (a) What will the following Python code print:

```
rom = {'i' : 1, 'v': 5, 'x' : 10}
s = "x+vii+xiv"
num = s.count('+')+1
print("There are", num, "items.")
rnums = s.split('+')
print("Last number is", rnums[-1])
print("First is", rom[rnums[0]])
sum = 0
for c in rnums[1]:
    sum = sum+rom[c]
    print(sum)
print("Middle is:", sum)
```

Answer:

```
There are 3 items.
Last number is xiv
First is 10
5
6
7
Middle is: 7
```

(b) The commands below are **run sequentially**, what is the output after each has run:

```
$ ls
nyc.csv  nyc.png  queens.png  si.png
$ pwd
/Users/csguest
```

setup.sh

```
echo "2V2V2V"
mkdir my_files
cd my_files
mkdir notes
mkdir work
```

i.
```
$ mv nyc.png man.png
$ ls
```

Answer:

```
man.png  nyc.csv  queens.png  si.png
```

ii.
```
$ mkdir images
$ mv *.png images
$ ls
```

Answer:

```
images  nyc.csv
```

iii.
```
$ mkdir nyc
$ ls nyc* | wc -l
```

Answer:

```
2
```

iv.
```
$ chmod +x setup.sh
$ ./setup.sh
$ cd my_files
$ ls
```

Answer:

```
2V2V2V
notes  work
```

2. (a) Check all that apply:

Answer:

i. What color is `tess` after this command? `tess.color("#FFFFFF")`
☐ black ☐ red ✓ white ☐ gray ☐ green

ii. Select all the **odd** binary numbers:
✓ 0001 ☐ 0100 ✓ 0111 ✓ 1011 ☐ 1110

iii. Select the hexadecimal numbers **larger than the decimal number 20**:
☐ 9 ☐ F ✓ 20 ✓ A6 ✓ FF

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

i.
```
nums = [ 'a', 'b', 'c', 'd', 'e', 'f']
```

Answer:

```
for i in range(  1, len(nums), 2  ):
    print(nums[i], end=" ")
```

Output:

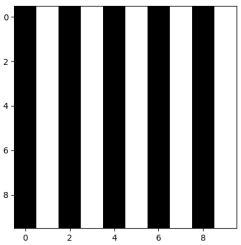
```
b d f
```

ii.

Answer:

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

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

Output:



(c) Consider the code:

Answer:

```
1  import turtle
2  tess = turtle.Turtle()
3  for i in range(5):
(i) 4      hex_col = input('Enter color (as hex): ")
5      tess.color(hex_col)
(ii) 6     tess forward(20)
7      tess.stamp()
```

i. **Circle** the code above and mark line with **(i)** that caused this error:

```
hex_col = input('Enter color (as hex): ")
                ^
SyntaxError: unterminated string literal (detected at line 4)
```

Write the code that would fix the error:

Answer:

```
hex_col = input('Enter color (as hex): ')
```

ii. **Box** the code above and mark line with **(ii)** that caused this error:

```
tess forward(20)
     ^^^^^^^
SyntaxError: invalid syntax
```

Write the code that would fix the error:

Answer:

```
tess.forward(20)
```

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

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

Answer:

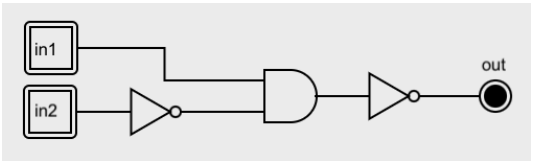
```
out = True
```

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

Answer:

```
out = False
```

iii.




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

Answer:

```
out = True
```

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

i.

```
in1 =  Answer: False
in2 =  Answer: True
```

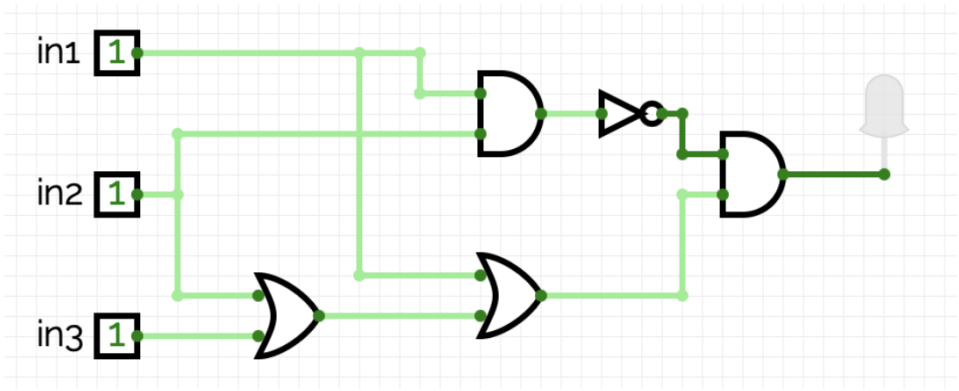
out = False

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

(c) Design a circuit that implements the logical expression:

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

Answer:



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

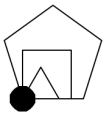
```
import turtle
tia = turtle.Turtle()
tia.shape("circle")

def ramble(t,side):
    if side < 3:
        t.stamp()
    else:
        for i in range(side):
            t.forward(side*10)
            t.left(360/side)
        ramble(t,side-1)
```

i. `ramble(tia,0)`

Answer:

ii. `ramble(tia,5)`

Answer:



(b) For the following code:

```
def helper(greet, name):
    if len(greet) < 5:
        greet = greet + greet
    msg = greet + " " + name
    return ( msg.lower() )
```

```
def v2():
    mess = "HeLLo"
    th = "Thomas"
    yun = helper(mess,th)
    return yun
```

i. What are the formal parameters for `helper()`:

Answer: greet, name

ii. What are the actual parameters for `helper()` when called in `v2()`:

Answer: mess, th

iii. What value does `v2()` return:

Answer: `"hello thomas"`

5. Design an algorithm that inputs two strings and returns true if, the two words use exactly the same letters (e.g. anagrams). For example, your algorithm should return false for `"monsoon", "moons"` since `"monsoon"` has 3 o's while `"moons"` has only 2. Your algorithm should return true for `"angered", "enraged"` since they have the same letters occurring the same number of times.

Answer:

Libraries:	`numpy`
Input:	two strings
Output:	True if anagrams, False otherwise

Design Pattern:

Answer: ☐ Accumulator ☐ Max/Min ✓ Finding Duplicates ☐ Searching

Principal Mechanisms (select all that apply):

Answer:

✓ 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.)

Answer:

(a) Ask the user for the two words.

(b) Set up an empty dictionary, `counts`.

(c) For each letter in the first word, If it's in the dictionary, increment its count. If not in the dictionary, set it's initial count to 1.

(d) For each letter in the second word, check if it's in the dictionary.

- If it is, decrement its value (and delete if value is now 0).
- If it's not in the dictionary, return False

(e) All letters matched, so, return True.

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

Answer:

```
#Mock Exam, S25, #7
import plotly.express as px
import pandas as pd
#Read in data:
name_str = input('Enter names, separated by spaces: ')
lat_str = input('Enter latitudes, separated by spaces: ')
lon_str = input('Enter longitudes, separated by spaces: ')
#Need to split up the inputted strings into lists:
names = name_str.split(' ')
lats = lat_str.split(' ')
lons = lon_str.split(' ')
#Set up a dictionary of the lists (used to make df):
data = {'latitude': lats, 'longitude': lons, 'name': names}
#Make a DataFrame of the dictionary:
df = pd.DataFrame(data)
#Use column names of df for keyword args:
```

```
fig = px.scatter_map(df,
                     lat="latitude",
                     lon="longitude",
                     hover_name="name")
#Save the output:
file_name = input('Enter output file name: ')
fig.write_html(file_name)
```

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: ")
first_names = input("Enter first: ")
emails = input("Enter email: ")
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) Consider the following MIPS program:

```
ADDI $s0, $zero, 2
ADDI $s1, $zero, 3
SUB $s2, $s1, $s0
ADD $s3, $s1, $s2
```

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

$s0 register	$s1 register	$s2 register	$s3 register
Answer: 2	**Answer:** 3	**Answer:** 1	**Answer:** 4

(b) Consider the MIPS code:

```
1  ADDI $sp, $sp, -5
2  ADDI $t0, $zero, 49
3  ADDI $s2, $zero, 57
4  SETUP: SB $t0, 0($sp)
5  ADDI $sp, $sp, 1
6  ADDI $t0, $t0, 2
7  BEQ $t0, $s2, DONE
8  J SETUP
9  DONE: ADDI $t0, $zero, 0
10 SB $t0, 0($sp)
11 ADDI $sp, $sp, -4
12 ADDI $v0, $zero, 4
13 ADDI $a0, $sp, 0
14 syscall
```

i) How many character are printed?	**Answer:** 4
ii) What is the first character printed?	**Answer:** 1
iii) What is the whole message printed?	**Answer:** 1 3 5 7
iv) Detail the changes needed to the code to print the message in reverse:	**Answer:** Line 2: Start t0 at 55. **Answer:** Line 3: Start s2 at 47. **Answer:** Line 6: Subtract 2 from t0.

9. (a) What is the output:

```
#include <iostream>
using namespace std;
int main()
{
    cout << "One fish, ";
    cout << "two fish" << endl << "red ";
    cout << "fish, blue fish\n";
    cout << "Dr. Seuss";
    return 0;
}
```

Answer:

```
One fish, two fish
red fish, blue fish
Dr. Seuss
```

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

```
#include <iostream>
using namespace std;
int main()
{
    int myst = -10, quest = 8;
    while ( (myst < 25) && (quest > 0) )
    {
        cout << myst << "\t" << quest << endl;

        myst += 10;
        quest -=2;

    }
    return 0;
}
```

Output:

-10	8
0	6
10	4
20	2

Answer:

(c) What is the output:

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

Answer:

```
#2#2#
2#2#2
#2#2#
2#2#2
#2#2#
```

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

Python program:

```python
num = 1
while num %2 != 0:
    num = int(input('Enter even number:'))
print("You entered", num)
```

C++ program:

Answer:

```cpp
#include <iostream>
using namespace std;
int main() {
    int num = 1;
    while (num % 2 != 0)
    {
        cout << "Enter even number: ";
        cin >> num;
    }
    cout << "You entered " << num;
    return 0;
}
```

(b) Write a **complete C++ program** that asks for the number of repetitions, print `"Hello!"` that many times.

A sample run of your code:

```
Enter repetition time:  5
Hello!
Hello!
Hello!
Hello!
```

```
Hello!
```

Answer:

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

int main()
{
    int reps;
    cout<< "Enter number of repetitions: ";
    cin >> reps;

    for(int i = 0; i < reps; i++)
    {
        cout<< "Hello!\n";
    }
    return(0);
}
```