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

May 2026

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

   i.
   ```
   s = "elion,gertrude;cohn,mildred;\
       petters,arlie"
   a = s[-5:]
   print(a.upper())
   ```

   Answer:
   ```
   ARLIE
   ```

   ii.
   ```
   names = s.split(';')
   print(names[-1])
   ```

   Answer:
   ```
   petters,arlie
   ```

   iii.
   ```
   firsts = [n[0] for n in names]
   print(firsts)
   ```

   Answer:
   ```
   ['e', 'c', 'p']
   ```

   iv.
   ```
    for n in names:
       w = n.split(',')
       print(w[1],w[0])
   ```

   Answer:
   ```
   gertrude elion
   mildred cohn
   arlie peters
   ```

   (b) Consider the following shell commands:

   ```
   $ ls
   cheese          pepperoni       peppers         pesto_sauce     tomato_sauce
   $ pwd
   /Users/pizza
   ```

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

i.
```
$ mkdir sauce
$ mv *_* sauce
$ ls
```

Answer:
```
cheese          pepperoni       peppers         sauce
```

ii.
```
$ cd sauce
$ echo "Sauce options:"
$ ls
```

Answer:
```
Sauce options:
pesto_sauce     tomato_sauce
```

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

Answer:
```
/Users/pizza
```

iv.
```
$ echo "Toppings:"
$ ls *e* | wc -l
```

Answer:
```
Toppings:
3
```

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

Answer:

i. Fill in the code below to change the turtle, `ryan`, to be the brightest green:

`ryan.color("#` | 0 | 0 | F | F | 0 | 0 | `")`

ii. Select all the **even binary numbers**:

✓ 1000 ☐ 1101 ☐ 0111 ✓ 1010 ☐ 1111

iii. Select all the **hexadecimal numbers larger than 16 decimal**:

☐ A ☐ B ✓ 90 ✓ 99 ✓ FF

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

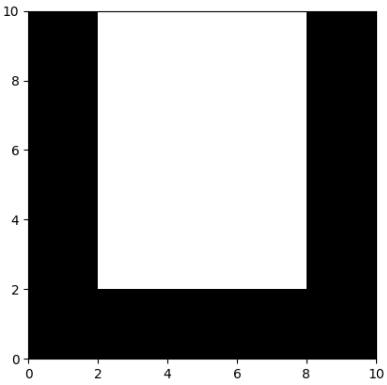
```
import matplotlib.pyplot as plt
import numpy as np
#Set background color to white:

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

logo [:, [      ],:]  = 0

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

Output:



Answer:

```
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[:,:2,:] = 0
logo[0:2,:,:] = 0
logo[:,-2:,:] = 0
plt.imshow(logo, extent= [0,10,0,10])
plt.show()
```

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

```
1 import turt
2 tia = Turtle.Turtle()
3 for i in range(5:
4     tia.right(360/5)
5     tia.forward(20)
```

i.
```
import turt
ModuleNotFoundError: No module named 'turt'
```

Answer: **Line Number:** 1 **Correct code:** `import turtle`

ii.
```
tia = Turtle.Turtle()
      ^^^^^^
NameError: name 'Turtle' is not defined.
```

Answer: **Line Number:** 2 **Correct code:** `tia = turtle.Turtle()`

iii.
```
for i in range(5:
                ^
SyntaxError: invalid syntax
```

Answer: | **Line Number:** 3 | **Correct code:** `for i in range(5):`

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

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

Answer:

`out = True`

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

`in1 =` _____

`in2 =` _____

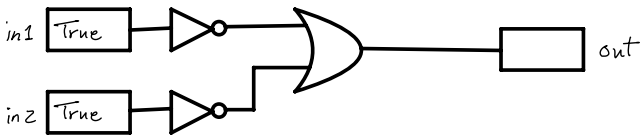
out = True

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

Answer:

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

(d) What is the output of this circuit?



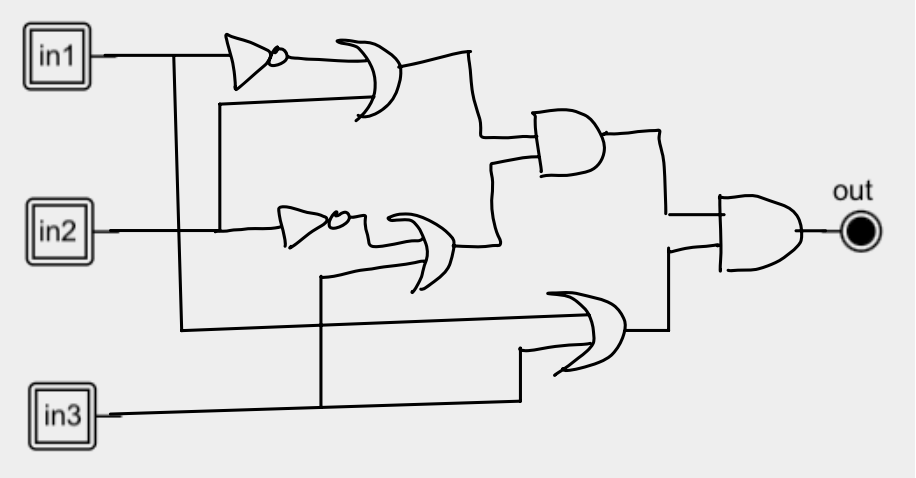
Answer:

`out = False`

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

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

V2: ((not in1 or in2) and (not in2 or in3)) and (in1 or in3)



Answer:

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

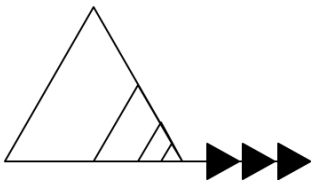
```
1  import turtle
2  tom = turtle.Turtle()
3  tom.shape('triangle')
4
5  def ramble(ty, dist, stamp):
6      if dist > 10:
7          for i in range(3):
8              ty.left(120)
9              ty.forward(dist)
10         ramble(ty,dist//2,stamp)
11     elif stamp:
12         for i in range(3):
13             ty.forward(20)
14             ty.stamp()
15     else:
16         ty.forward(20)
```

i. `ramble(tom,8,False)`

Answer:

ii. `ramble(tom,100,True)`

Answer:



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

Answer: ty, dist, stamp

(c) If you call `ramble(tom,8,False)`, which branches of the function are tested (check all that apply):

Answer:

- ☐ The block of code at Lines 7-10.
- ☐ The block of code at Lines 12-14.
- ✓ The block of code at Line 16.
- ☐ None of these blocks of code (lines 7-10, 12-14, 16) are visited from this invocation (call).

(d) If you call `ramble(tom,100,True)`, which branches of the function are tested (check all that apply):

Answer:

- ✓ The block of code at Lines 7-10.
- ✓ The block of code at Lines 12-14.
- ☐ The block of code at Line 16.
- ☐ None of these blocks of code (lines 7-10, 12-14, 16) are visited from this invocation (call).

5. Design an algorithm that takes a string and returns the most common word in the string. If there is not a unique word that occurs most often, return the first one alphabetically. Your algorithm, if given the input:

```
"A rose is a rose is a rose.  -Gertrude Stein"
```

would return since **a** since `a`, `rose`, and `is` all occur 3 times but `a` is first alphabetically.

Answer:

Libraries:	none
Input:	a string containing words separated by spaces
Output:	the word that occurs most often

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) Input the string from the user.
(b) Use `.lower()` to convert the string to lower case.
(c) Split the string into words
(d) Set up an empty dictionary, `new_dict`.
(e) For word in the word list:
(f) Check if the word is in the dictionary.
(g) If it is, increment the count
(h) If it isn't, add word with value 1 to the dictionary.
(i) Find the maximum value in the dictionary and return its key. If there's ties, return the first alphabetically.

6. Fill in the Python program that will:

   - prompt the user for the name of a CSV file,
   - prompt the user for the name of a column in that CSV file,
   - create a new column that is double the column that the user provided,
   - print out the mean value of the double column, and
   - displays a plot of the double column (with `"Year"` as the x-axis).

   Answer:

```
#Import the libraries for data frames and displaying images as pd and plt:
import pandas as pd
import matplotlib.pyplot as plt

#Prompt user for file name:
file_name = input("Enter the name of the CSV file: ")

#Prompt user for column name:
col_name = input("Enter the name of a column: ")

#Read in the CSV file to a DataFrame:
df = pd.read_csv(file_name)

#Create a new column that's double that of the column specified:
df["Double"] = 2 * df[col_name]

#Compute the average value of the column, "Double":
m = df["Double"].mean()
print("Mean of column 'Double' is", m)

#Display a plot of "Year" vs. "Double" column
df.plot(x="Year", y="Double")
plt.show()
```

7. Write a **complete program** that identifies all two-part numbers from a string and returns a list of those numbers. In the input string, each number is separated by a comma followed by a space: `', '`, and two-part numbers contain a `'-'`.

   For example, if you ran your program and the user entered:

   `Enter numbers, separated by commas: twenty, four, thirty-one, sixty-six, eight`

   A list of the two-part numbers printed is: `["thirty-one", "sixty-six"]`.

   Answer:

```
numbers = input("Enter numbers, separated by commas: ")
```

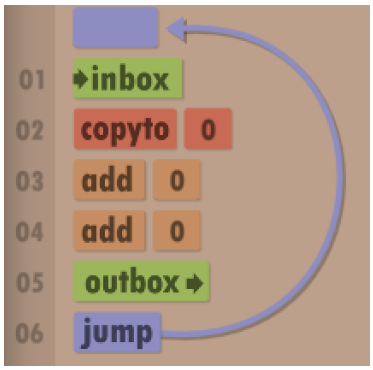
```
nums = numbers.split(", ")

two_part = []

for num in nums:
    if "-" in num:
        two_part.append(num)

print("A list of the two-part numbers:", two_part)
```

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



Answer: 18, -12, 15, 0

(b) Consider the following MIPS program:

```
ADDI $s0, $zero, 10
ADDI $s1, $s0, 5
ADD $s2, $s1, $s1
SUB $s3, $s0, $s2
```

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

$s0 register	$s1 register	$s2 register	$s3 register
Answer: 10	**Answer:** 5	**Answer:** 10	**Answer:** 0

(c) Consider the MIPS code:

```
1  ADDI $sp, $sp, -9
2  ADDI $t0, $zero, 33
3  ADDI $s2, $zero, 8
4  SETUP: SB $t0, 0($sp)
5  ADDI $sp, $sp, 1
6  ADDI $s2, $s2, -1
7  BEQ $s2, $zero, DONE
8  J SETUP
9  DONE: ADDI $t0, $zero, 0
10 SB $t0, 0($sp)
11 ADDI $sp, $sp, -8
12 ADDI $v0, $zero, 4
```

```
13 ADDI $a0, $sp, 0
14 syscall
```

message printed?	
iv) List what you need to change to print half of the message:	* line 1: Set $sp to -5. * line 3: Set the counter $s2 to 4. * line 11: Set $sp to -4.

Answer:

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

```
//Quote by George R.R. Martin
#include <iostream>
using namespace std;
int main()
{
    cout << "A mind needs books ";
    
    cout << "a whetstone," << endl;
    cout << "if it is to keep its edge.";
    return 0;
}
```

Output:

A mind needs books as
a sword needs a whetstone,
if it is to keep its edge.

Answer:

```
    cout << "as \na sword needs ";
```

(b) What is the output:

```
//More Game of Thrones
#include <iostream>
using namespace std;
int main()
{
    int count = 3;
    while (count > 1) {
        cout <<"Winter is coming ";
        count--;
    }
    cout << "!\nNothing burns ";
    cout << "like the cold." << endl;
    return 0;
}
```

Answer:

```
Winter is coming Winter is coming !
Nothing burns like the cold.
```

(c) What is the output:

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

Answer:

```
0aaaa
1bbbb
2aaaa
3bbbb
4aaaa
```

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

C++ program:

```cpp
#include <iostream>
using namespace std;
int main()
{
    float fah;
    cout<<"Enter temperature in fahrenheit: ";
    cin >> fah;
    float cel = ((5.0/9.0)* (fah - 32));
    cout << cel << endl;
    return 0;
}
```

Python program:

Answer:

```python
fah = float(input("Enter temperature in fahrenhe
cel = ((5.0/9.0)* (fah - 32))
print(cel)
```

(b) Write a **complete C++ program** that asks for a positive whole number, `num`, and prints out the partial sum up to `num` (e.g. 1, $1+2$, $1+2+3$,...,$1+2+...+num$).

A sample run of your code:

```
Enter num:  5
1
3
6
10
15
```

Answer:

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

int main()
{
    int num;
    int sum=0;
    cout<< "Enter number: ";
    cin >> num;

    for(int i = 1; i <= num; i++)
    {
        sum = sum + i;
        cout<< sum << endl;
    }
    return(0);
}
```