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

Final Examination, Version 1
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 = "v+xvi+xx"
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 xx
First is 5
10
15
16
Middle is: 16
```

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

```
$ ls
hello.cpp  p1.py  pp_hello.py  setup.sh
$ pwd
/Users/csguest
```

setup.sh

```
echo "Setting up"
mkdir project
cd project
mkdir programs
mkdir data
```

i.
```
$ mv hello.cpp   p1.cpp
$ ls
```

Answer:

```
p1.cpp          p1.py          pp_hello.py          setup.sh
```

ii.
```
$ mkdir cprogs
$ mv *.cpp  cprogs
$ ls
```

Answer:

```
cprogs  pp_hello.py  setup.sh
```

iii.
```
$ mkdir pp_5
$ ls | grep pp
```

Answer:

```
pp_5
```

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

Answer:

```
Setting up
program  data
```

2. (a) Check all that apply:

Answer:

i. What color is `tess` after this command? `tess.color("#FF0000")`
☐ black ✓ red ☐ white ☐ gray ☐ green

ii. Select all the **odd** binary numbers:
☐ 0000 ✓ 0101 ✓ 0111 ☐ 1010 ✓ 1111

iii. Select the hexadecimal numbers **smaller than the decimal number 20**:
✓ A ✓ 11 ☐ 20 ☐ 23 ☐ 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, -4, -1  ):
    print(nums[i], end=" ")
```

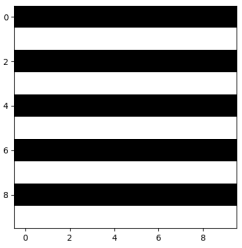
Output:

```
f e d
```

Answer:

ii.
```
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:

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

Answer:

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

```
    for i in range(5)
                      ^
SyntaxError: expected ':'
```

Write the code that would fix the error:

Answer:

```
for i in range(5):
```

ii. **Box** the code above and mark line with **(ii)** 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): ')
```

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

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

Answer:

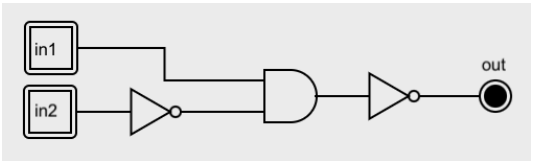
```
out = False
```

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

Answer:

```
out = False
```

iii.




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

Answer:

```
out = True
```

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

i.

```
in1 =
```
Answer: `True`

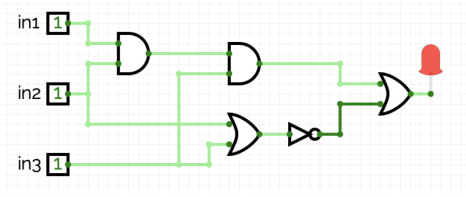
```
in2 =
```
Answer: `False`

out = `False`

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

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

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



Answer:

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

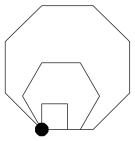
```
import turtle
tom = turtle.Turtle()
tom.shape("circle")

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-2)
```

i. `ramble(tom,0)`

Answer:

ii. `ramble(tom,8)`

Answer:



(b) For the following code:

```
def helper(calvin, reps):
    if reps <= 0:
        return "NA"
    else:
        return (calvin*reps)
```

```
def v1():
    wafa = "a"
    faye = 5
    yun = helper(wafa,faye)
    return yun
```

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

Answer: calvin, reps

ii. What are the actual parameters for `helper()` when called in `v1()`:

Answer: None

iii. What value does `v1()` return:

Answer: aaaaa

5. Design an algorithm that inputs a string and returns true if, after removing spaces and punctuation, its the same written forward and backwards (e.g. a palindrome). For example, your algorithm should return false for `"Hello Mom!"` since `"hellomom"` and `"momolleh"` are not the same, but would return true for `"A man, a plan, a canal, Panama."` since `"amanaplanacanalpanama"` is the same forward and backwards.

Answer:

Libraries:	`none`
Input:	a string
Output:	Boolean (True/False) value

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 user for input string and store it in variable.

(b) Remove spaces and punctuation from the string (either by looping or by using the string function `replace(" ", "")`).

(c) Make a copy of the string reversed, either by looping through the original string backwards (from str[-1] to str[0]) and append characters to reversed string, or by slicing backwards `reversed = str[-1::-1]`.

(d) Use if-else statement to compare input string to reversed string.

(e) Or, instead of making a second copy of the string, loop backward/forward through the string at the same time.

(f) If input string is the same as reversed string, output a message saying input string is a palindrome.

(g) Else, output a message saying input string is not a palindrome.

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: ")
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, 3
ADD $s1, $s0, $s0
ADD $s2, $s1, $s0
SUB $s3, $s1, $s2
```

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

$s0 register	$s1 register	$s2 register	$s3 register
Answer: 3	**Answer:** 6	**Answer:** 9	**Answer:** -3

(b) Consider the MIPS code:

```
1  ADDI $sp, $sp, -5
2  ADDI $t0, $zero, 50
3  ADDI $s2, $zero, 58
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 characters are printed?	**Answer:** 4
ii) What is the first character printed?	**Answer:** 2
iii) What is the whole message printed?	**Answer:** 2468
iv) Detail the changes needed to the code to print the message in reverse:	**Answer:** Line 2: Start t0 at 56. **Answer:** Line 3: Start s2 at 48. **Answer:** Line 6: Subtract 2 from t0.

9. (a) What is the output:

```
#include <iostream>
using namespace std;
int main()
{
    cout << "Twinkle, twinkle, little";
    cout << "star,\nhow I wonder";
    cout << "what you are"<<endl<<"Jane";
    cout << "Taylor";
    return 0;
}
```

Answer:

```
Twinkle, twinkle, little star,
How I wonder what you are
Jane Taylor
```

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

```
#include <iostream>
using namespace std;
int main()
{
    int myst = -5, quest = 5;
    while ( (myst < 15) && (quest > 0) )
    {
        cout << myst << "\t" << quest << endl;
```

Output:

-5	5
0	4
5	3
10	2

Answer:

```
        myst += 5;
        quest--;

    }
    return 0;
}
```

(c) What is the output:

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

Answer:

```
@
11
@@@
1111
@@@@@
```

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

Python program:

```
year = 0
while year <= 2026:
    year = int(input('Enter grad year: '))
print("You entered", year)
```

C++ program:

Answer:

```
#include <iostream>
using namespace std;
int main() {
    int year = 0;
    while (year <= 2026)
    {
        cout << "Enter grad year: ";
        cin >> year;
    }
    cout << "You entered " << year;
    return 0;
}
```

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

A sample run of your code:

```
Enter repetition time:  5
Practice makes perfect.
Practice makes perfect.
Practice makes perfect.
Practice makes perfect.
Practice makes perfect.
```

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<< "Pratice!\n";
    }
    return(0);
}
```