Answer Key:

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

Spring 2025

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

```
num_s = "two twenty-one thirty-two three twenty"
nums = num_s.split(" ")
print(nums[-1])
count = num_s.count("-")
print("List has", count, "two-parts.")
two_nums = [n for n in nums if "-" in n]
print(two_nums)
ones = ["zero","one","two","three","four"]
tens = ["","","twenty","thirty","forty"]
for num in two_nums:
    places = num.split("-")
    dec = ones.index(places[1])+\
          tens.index(places[0])*10
    print(num, "=", dec)
```

Answer Key:

```
List has 2 two-parts.
twenty
['twenty-one', 'thirty-two']
twenty-one = 21
thirty-two = 32
```

(b) Consider the following shell commands:

```
$ ls
code_p1.png        code_p2.png        exams              key.pdf
$ file exams
exams: directory
$ pwd
/tmp/final/ver2
```

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

i.
```
$ mv key.pdf exams
$ ls
```

Answer Key:

```
code_p1.png        code_p2.png        exams
```

ii.
```
$ cd exams
$ cp key.pdf key_2.pdf
$ ls
```

Answer Key:

```
key_2.pdf        key.pdf
```

iii.
```
$ mkdir answers
$ cd answers
$ pwd
```

Answer Key:

```
/tmp/final/ver2/exams/answers
```

iv.
```
$ cd ../../
$ ls | grep c
```

Answer Key:

```
code_p1.png
code_p2.png
```

2. (a) Fill in the missing values in the table:

Answer Key:

Decimal	Binary	Hexadecimal
5	**101**	5
12	1100	C
33	100001	**21**
253	11111101	**FD**

(b) Fill in the missing code to make the image:

(c) Consider the code:

Answer Key:

```
1  import pandas as pd
2  csvFile = input('Enter CSV file name: ')
3  recipe = pds.read_csv(csvFile)
4  recipe["Amount'] = 2*recipe["Amount"]
5  print(recipe)
```

(i) (ii)

The answer should include:

- Mark line 3 with a "(i)".
- In line 3, circle the word `pds`.
- Mark line 4 with a "(ii)".
- On line 4, should box the single quote (') that does not match the starting double quote.

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

```
line 3: recipe = pds.read_csv(csvFile)
                 ^^^
NameError: name 'pds' is not defined. Did you mean: 'pd'?
```

Write the code that would fix the error:

Answer Key:

```
recipe = pd.read_csv(csvFile)
```

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

```
line 4: recipe["Amount'] = 2*recipe["Amount"]
                                              ^
SyntaxError: unterminated string literal (detected at line 4)
```

Write the code that would fix the error:

Answer Key:

```
recipe["Amount"] = 2*recipe["Amount"]
```

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

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

Answer Key:

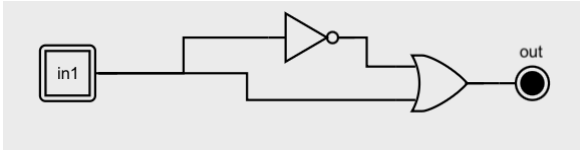
```
out = True
```

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

Answer Key:

```
out = True
```

iii.




```
in1 = False
```

Answer Key:

```
out = True
```

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

i.

in1 = **Answer Key:** False

in2 = **Answer Key:** True

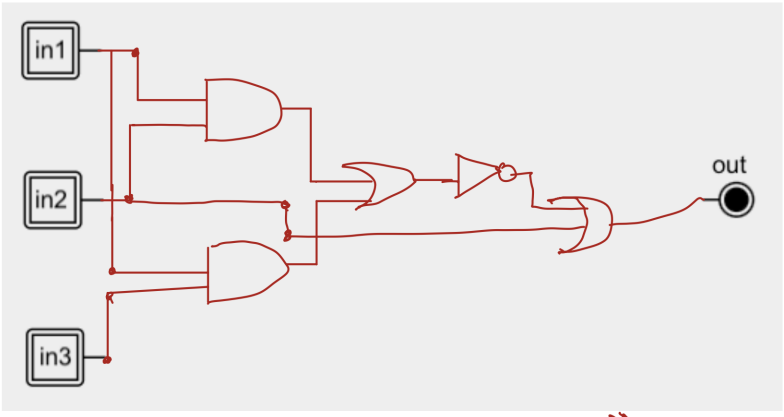
out = False

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

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

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

Answer Key:



v2: not((in1 and in2) or (in1 and in2)) or in2

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

i. `ramble(tiago,0)`

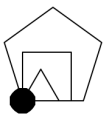
Answer Key:

```
import turtle
tiago = turtle.Turtle()
tiago.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)
```

ii. `ramble(tiago,5)`

Answer Key:



(b) For the following code:

```
def start():
    gwenael = 8
    karen = 10
    ryan = v2(gwenael,karen)
    return ryan
```

```
def v2(jaime, lily):
    if jaime + lily > 10:
        return lily
    else:
        return -1
```

i. What are the formal parameters for `v2()`:

Answer Key: jaime, lily

ii. What are the formal parameters for `start()`:

Answer Key: None

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

Answer Key: 10

5. Write a function `most_common()` that takes a string, converts it to lower case, and returns the character that occurs most in the string. If there is a tie for most occurrences, return the first alphabetically. For example:

```
most_common("Mihi cura futuri")
```

would return `i` since both `i` and `u` occur the most times (3), but `i` is first alphabetically.

Answer Key:

Libraries:	No additional– just core Python
Input:	a string
Output:	the character that occurs most often

Design Pattern:

Answer Key:

☐ Accumulator ✓ Max/Min ✓ Finding Duplicates ☐ Searching

Principal Mechanisms (select all that apply):

Answer Key:

✓ 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 Key:

(a) Set up an empty dictionary, `new_dict`.

(b) Use `.lower()` to convert the string to lower case.

(c) For character in the string:

(d) Check if the character is in the dictionary.

(e) If it is, increment the count

(f) If it isn't, add ID with value 1 to the dictionary.

(g) Find the maximum value in the dictionary and return its key.

6. Fill in for the code below to create an interactive map, based on housing data. Your program should ask the user for the input and output file names. It should read in the CSV file and create a new column that sums up the number of studio, 1-bedroom, and 2+ bedroom apartments in a single new column, `Total Units`. A interactive HTML map, based on the DataFrame entries, is saved to the specified outfile.

Answer Key:

```
#Import pandas and plotly express libraries:
import plotly.express as px
import pandas as pd
#Ask user for file name:
file_name = input('Enter names, separated by spaces: ')
#Read in the file to a DataFrame:
df = pd.read_csv(file_name)
#Make a new column that sums up "Studio", "1Bed", "2+Bed" columns:
df["Total Units"] = df["Studio"] + df["1Bed"] + df["2+Bed"]
#Use df to make a scatter_map: columns: "latitude" and "longitude" for location,
# "Project Name" for hover_name, & "Total Units" for size:
fig = px.scatter_map(df,
                     lat="latitude",
                     lon="longitude",
                     hover_name="name",
                     size="Total Units")
#Ask user for output file name:
html_file = input('Enter output file name: ')
#Save the file to html:
fig.write_html(html_file)
```

7. Write a **complete Python program** that

- asks the user for the name of a png file and
- prints the number of pixels that are bright red (the fraction of red is above 0.75 and the fraction of green, and the fraction of blue are below 0.25).

Answer Key:

```
#Count number of red pixels in an image

#Import the packages for images and arrays:
```

```
import matplotlib.pyplot as plt
import numpy as np

fileName = input('Enter file name: ')
img = plt.imread(fileName) #Read in image
countRed = 0               #Number of pixels that are bright red

#For every pixel:
for i in range(img.shape[0]):
     for j in range(img.shape[1]):
          if (img[i,j,0] > 0.75) and (img[i,j,1] < 0.25) and (img[i,j,2] < 0.25):
               countRed = countRed + 1

print("Red count is", countRed)
```

8. (a) Consider the following MIPS program:

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

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

$s1 register	$s2 register	$s3 register
Answer Key: 2	**Answer Key:** 3	**Answer Key:** 4

(b) Consider the MIPS code:

```
1  ADDI $sp, $sp, -4
2  ADDI $t0, $zero, 83
3  ADDI $s2, $zero, 86
4  SETUP: SB $t0, 0($sp)
5  ADDI $sp, $sp, 1
6  ADDI $t0, $t0, 1
7  BEQ $t0, $s2, DONE
8  J SETUP
9  DONE: ADDI $t0, $zero, 0
10 SB $t0, 0($sp)
11 ADDI $sp, $sp, -3
12 ADDI $v0, $zero, 4
13 ADDI $a0, $sp, 0
14 syscall
```

Answer Key:

i) How many characters are printed?	**3**
ii) What is the first character printed?	**S**
iii) What is the whole message printed?	**STU**
iv) Detail the changes needed to the code to print the message in reverse:	**Line 2: Start t0 at 85.** **Line 3: Start s2 at 82.** **Line 6: Subtract 1 from t0.**

9. (a) What is the output

```
//Neil deGrasse Tyson
#include <iostream>
using namespace std;
int main()
{
    cout << "There is no "
         << "greater educ";
    cout << "ation\nthan one ";
    cout << "that is self-driven."
         << endl;
}
```

Answer Key:

```
There is no greater education
than one that is self-driven.
```

(b) What is the output:

```
#include <iostream>
using namespace std;
int main()
{
  cout << "Begin" << endl;
  int x = 2;
  while (x > 0) {
    cout <<"Again\n";
    x--;
  }
  cout << "End" << endl;
  return 0;
}
```

Answer Key:

```
Begin
Again
Again
End
```

(c) What is the output:

```cpp
#include <iostream>
using namespace std;
int main(){
    for (int i=0; i<4; i++){
        for(int j=0; j<4; j++){
            if ( j % 2 == 0)
                cout<<"+";
            else
                 cout<<"-";
         }
         cout << endl;
     }

      return 0;
}
```

Answer Key:

```
+-+-
+-+-
+-+-
+-+-
```

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

Python program:

```python
num = 1
while (num > 100) or (num % 2 == 1):
    num = int(input("Enter small even #: "))
print("Your number:", num)
```

C++ program:

Answer Key:

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

(b) Write a C++ program that will ask for the time in 24 hour format (e.g. 2034 is 8:34pm) and, prints out "Good Morning" if it is before noon (e.g. 1200), "Good Evening" if it after 6pm (e.g. 1800), and otherwise print "Good Afternoon."

A sample run:

```
Enter time:  1345
Good Afternoon
```

Answer Key:

```cpp
#include <iostream>
using namespace std;
int main()
{
  int time;
  cout << "Enter time: ";
  cin >> time;
  if (time < 1200) {
    cout << "Good Morning \n";
  }
  else if (time > 1800){
    cout << "Good Evening \n";
  }
  else {
    cout << "Good Afternoon \n";
  }
  return 0;
}
```