

Answer Key:

FINAL EXAM, VERSION 3
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 = "one,twenty,thirty three,four,twenty two"
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-part numbers.  
['thirty three', 'twenty two']  
thirty three = 33  
twenty two = 22

- (b) Consider the following shell commands:

```
$ ls
p1_out.png p10_out.png images lect1.pdf
$ file images
images: directory
$ pwd
/tmp/final/ver3
```

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

- i. \$ mv lect1.pdf l1.pdf  
\$ ls

**Answer Key:**

images

l1.pdf

p1\_out.png

p2\_out.png

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

**Answer Key:**

- images      l1.pdf  
 iii. `$ cd images`  
`$ pwd`

**Answer Key:**

/tmp/final/ver3/images

- iv. `$ echo "Num is:"`  
`$ ls p* | wc -l`

**Answer Key:**

Num is:  
 2

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

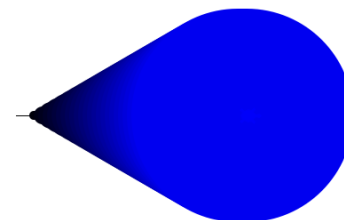
| Decimal   | Binary     | Hexadecimal |
|-----------|------------|-------------|
| 5         | <b>101</b> | 5           |
| <b>12</b> | 1100       | C           |
| 33        | 100001     | <b>21</b>   |
| 253       | 11111101   | <b>FD</b>   |

**Answer Key:**

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

**Answer Key:**

```
import turtle
turtle.colormode(255)
tia = turtle.Turtle()
tia.shape("turtle")
tia.backward(100)
#For 0,10,20,...,250
for i in range(0,255,10):
 tia.forward(10)
 tia.pensize(i)
 tia.color(0,0,i)
```



- (c) Consider the code:

**Answer Key:**

```

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

```

The answer should include:

- Mark line 2 with a “(i)”.
- On line 2, should circle the single quote (') that does not match the starting double quote.
- Mark line 3 with a “(ii)”.
- In line 3, circle the word pds.

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

```
csvFile = input("Enter CSV file name: '')
```

SyntaxError: unterminated string literal (detected at line 2)

Write the code that would fix the error:

#### Answer Key:

```
csvFile = input("Enter CSV file name: ")
```

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

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

```
in1 = True
```

i. in2 = True

```
out = in1 or in2
```

#### Answer Key:

```
out = True
```

```
in1 = True
```

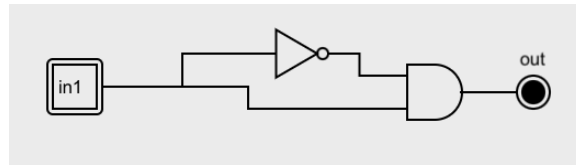
ii. in2 = False

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

#### Answer Key:

```
out = False
```

iii.



in1 = False

**Answer Key:**

out = False

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

i. 

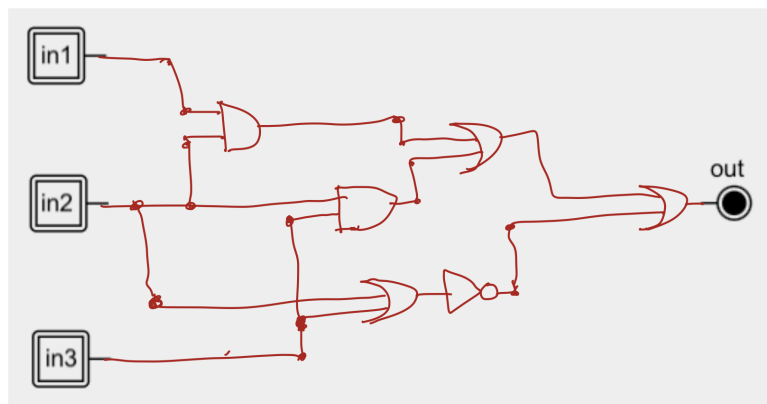
|                         |
|-------------------------|
| <b>Answer Key:</b> True |
| <b>Answer Key:</b> True |

out = in1 and (not in1 or in2)

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

$((in1 \text{ and } in2) \text{ or } (in2 \text{ and } in3)) \text{ or not } (in2 \text{ or } in3)$

**Answer Key:**



v3:  $((in1 \text{ and } in2) \text{ or } (in2 \text{ and } in3)) \text{ or not } (in2 \text{ or } in3)$

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

i. `ramble(tia,0)`

**Answer Key:**

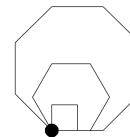
```
import turtle
tia = turtle.Turtle()
tia.shape("circle")

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-2)
```



ii. `ramble(tia,8)`

**Answer Key:**



(b) For the following code:

```
def v1(vincent, munem):
 if vincent + munem > 0:
 return vincent
 else:
 return -1
```

```
def start():
 panda = 20
 minh = -30
 qiuqun = v1(panda,minh)
 return qiuqun
```

i. What are the formal parameters for `v1()`:

**Answer Key:** vincent, munem

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

**Answer Key:** None

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

**Answer Key:** 20

5. Write a function `most_frequent()` that takes a list of 8-digit strings and returns the string that occurs most in the list. If there is a tie for most occurrences, return the first alphabetically. For example:

```
ids = ['12345678', '11223344', '12312323', '12345678']
```

```
most_frequent(ids)
```

would return '12345678' since it occurs twice, while all other entries once.

|                    |                   |                                 |
|--------------------|-------------------|---------------------------------|
| <b>Answer Key:</b> | <b>Libraries:</b> | No additional– just core Python |
|                    | <b>Input:</b>     | list of 8-digit strings         |
|                    | <b>Output:</b>    | the ID 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) For each ID in the ID list:
  - (c)    Check if the ID is in the dictionary.
  - (d)    If it is, increment the count
  - (e)    If it isn't, add ID with value 1 to the dictionary.
  - (f) 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 blue (the fraction of blue is above 0.75 and the fraction of green, and the fraction of red are below 0.25).

### Answer Key:

```

#Count number of blue 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
countBlue = 0 #Number of pixels that are bright blue

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

print("Blue count is", countBlue)

```

8. (a) Consider the following MIPS program:

```

ADDI $s0, $zero, 4
ADD $s1, $s0, $s0
ADD $s2, $s1, $s1
SUB $s3, $s2, $s0

```

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

| \$s1 register        | \$s2 register         | \$s3 register         |
|----------------------|-----------------------|-----------------------|
| <b>Answer Key:</b> 8 | <b>Answer Key:</b> 16 | <b>Answer Key:</b> 12 |

(b) Consider the MIPS code:

```

1 ADDI $sp, $sp, -4
2 ADDI $t0, $zero, 68
3 ADDI $s2, $zero, 71
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?                                        | <b>3</b>                                                                                   |
| ii) What is the first character printed?                                   | <b>D</b>                                                                                   |
| iii) What is the whole message printed?                                    | <b>DEF</b>                                                                                 |
| iv) Detail the changes needed to the code to print the message in reverse: | <b>Line 2: Start t0 at 70.<br/>Line 3: Start s2 at 67.<br/>Line 6: Subtract 1 from t0.</b> |

9. (a) What is the output



```
//Derek Bok
#include <iostream>
using namespace std;
int main()
{
 cout << "If you think education";
 cout << endl << "is expensive";
 cout << "ive,\nTry ignorance.\n";
 return 0;
}
```

**Answer Key:**

If you think education  
is expensive,  
Try ignorance.

(b) What is the output:

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

**Answer Key:**

Hi  
:  
:  
Bye

(c) What is the output:

```

#include <iostream>
using namespace std;
int main(){
 for (int i=0; i<4; i++){
 for(int j=0; j<4; j++){
 if (i % 2 == 0)
 cout<<"0";
 else
 cout<<"1";
 }
 cout << endl;
 }

 return 0;
}

```

### Answer Key:

```

0000
1111
0000
1111

```

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

C++ program:

### Answer Key:

```

#include <iostream>
using namespace std;
int main()
{
 int num = 1;
 while ((num < 100) || (num%2 == 0))
 {
 cout << "Enter large odd #:";
 cin >> num;
 }
 cout << "Your number: " << num;
 return 0;
}

```

Python program:

```

num = 1
while (num < 100) or (num % 2 == 0):
 num = int(input("Enter large odd #: "))
print("Your number:", num)

```

- (b) Write a C++ program that will ask for the time in 24 hour format (e.g. 2034 is 8:34pm) and, prints out “Early” if it is before noon (e.g. 1000), “Late” if it after 7pm (e.g. 1900), and otherwise print “Just Right.”

A sample run:

```

Enter time: 1345
Good Afternoon

```

### Answer Key:

```
#include <iostream>
using namespace std;
int main()
{
 int time;
 cout << "Enter time: ";
 cin >> time;
 if (time < 1000) {
 cout << "Early \n";
 }
 else if (time > 1900){
 cout << "Late \n";
 }
 else {
 cout << "Just Right \n";
 }
 return 0;
}
```