

Answer Key:

FINAL EXAM, VERSION 3
CSci 127: Introduction to Computer Science
Hunter College, City University of New York

20 December 2021

1. (a) Given the quote in the code below, fill in the code to produce the Output on the right:

```
quote = ' "My life is my message." -- Mahatma Gandhi'
```

i. `print(quote[])`

Answer Key:

-14:

ii. `print(quote[2:4].)`

Answer Key:

`lower()`

iii. `print("This quote has", end=" ")`
`print(quote.count() , "period")`

Answer Key:

`count('.')`

- (b) Fill in the code below to produce the Output on the right:

```
numbers = "10 / 11 / 12 / 13 / 14"
```

i. `num_list = numbers.`

Answer Key:

`split(' / ')`

ii. `for n in num_list :`
`print()`

Answer Key:

```
int(n)%2
```

(c) Consider the following shell commands:

```
$ ls
avg.py images logarithm.py logo.png reds.png
```

- i. What is the output for:
`$ ls | grep log`

Answer Key:

```
logarithm.py      logo.png
```

- ii. What is the output for:
`$ mkdir images/colors`
`$ mv logo.png images`
`$ mv reds.png images/colors`
`$ cd images`
`$ ls`

Answer Key:

```
colors logo.png
```

- iii. What is the output for:
`$ cd ../`
`$ ls *.py`

Answer Key:

```
avg.py logarithm.py
```

2. (a) Select the color corresponding to the rgb values below:

Answer Key:

- i. `rgb = (255, 0, 255)`
☐ black ☐ red ☐ white ☐ gray ☒ purple
- ii. `rgb = "#000000"`
☒ black ☐ red ☐ white ☐ gray ☐ purple

- iii. `rgb = (0.5, 0.5, 0.5)`
☐ black ☐ red ☐ white ☒ gray ☐ purple
- iv. Select the SMALLEST Binary number:
☐ 110100 ☐ 011101 ☐ 101000 ☒ 000111 ☐ 101010
- v. What is the Hexadecimal number equivalent to decimal 15?
☒ 0F ☐ 99 ☐ A0 ☐ FF ☐ C3
- (b) Given the list `names` below, fill in the code to produce the Output on the right:
- ```
names = ["aida", "mandy", "diana", "roziena", "nancy", "lola"]
```

i. **Answer Key:**

```
for i in range(3):
 print(names[i], end=" ")
```

**Output:**

```
aida mandy diana
```

ii. **Answer Key:**

```
for j in range(1, 6, 2):
 print(names[j], end=" ")
```

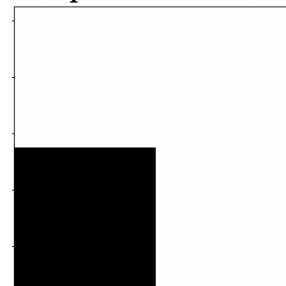
**Output:**

```
mandy roziena lola
```

**Answer Key:**

```
import numpy as np
iii. import matplotlib.pyplot as plt
im = np.ones((10,10,3))
im[5 : , 5 : , :] = 0
plt.imshow(im)
plt.show()
```

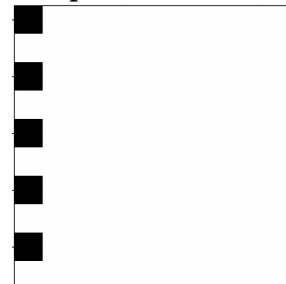
**Output:**



**Answer Key:**

```
import numpy as np
iv. import matplotlib.pyplot as plt
im = np.ones((10,10,3))
im[: : 2 , : 1 , :] = 0
plt.imshow(im)
plt.show()
```

**Output:**



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

```

in1 = True
i. in2 = False
 out = not in1 and in2

```

**Answer Key:**

```
out = False
```

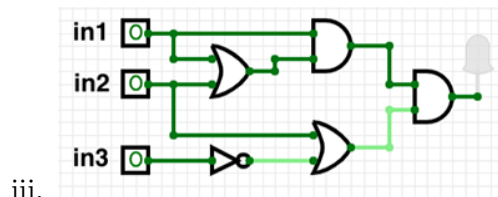
```

in1 = True
ii. in2 = False
 in3 = in1 or not in2
 out = not(in1 or not in2) and not in3

```

**Answer Key:**

```
out = False
```



```

in1 = True
in2 = False
in3 = False

```

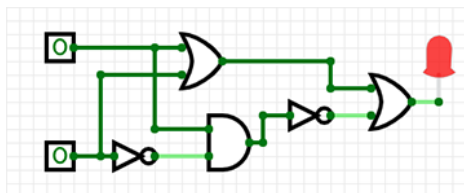
**Answer Key:**

```
out = True
```

(b) Draw a circuit that implements the logical expression:

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

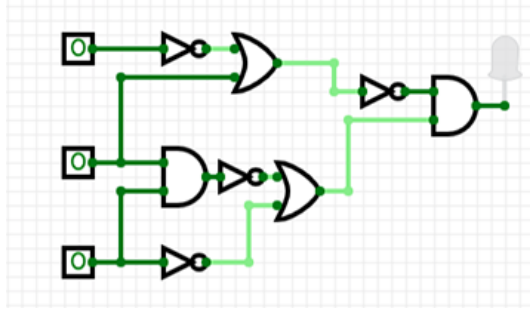
**Answer Key:**



(c) Fill in the circuit with the gate-symbol or gate-name that implements the logical expression:

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

**Answer Key:**



4. Consider the following functions:

```
def count(items, ch):
 count = 0
 for i in range(len(items)):
 if compare(items[i], ch):
 count += 1
 return count
```

```
def compare(i, c):
 return i == c
```

```
def main():
 chars = ['a', 'a', 'b', 'c', 'b', 'b', 'c', 'c']
 print(count(chars, 'a'))
```

(a) What are the formal parameters for `compare()`?

**Answer Key:** `i, c`

(b) What are the actual parameters for `count()`?

**Answer Key:** `chars, 'a'`

(c) How many calls are made to `compare()` after calling `main()`?

**Answer Key:** 8

(d) What is the output after calling `main()`?

**Output:**

**Answer Key:**

2

5. Design an algorithm that asks the user for the name of a text file containing a grid of numbers and loads it into a 2D array of integers (think like an image without the color channel), as well as an input number `n`. The program outputs the index (`row, col`) of the first occurrence of `n` in the array.

**Libraries:**

**Answer Key:** numpy

**Input:**

**Answer Key:** The input file and number n

**Output:**

**Answer Key:** The index of n

**Design Pattern:**

**Answer Key:** ☒ Search      ☐ Find Min      ☐ Find Max      ☐ Find All

**Principal Mechanisms (select all that apply):**

**Answer Key:** ☐ Search      ☐ Single Loop      ☒ Nested Loop      ☒ Conditional (if/else) statement

☒ Indexing / Slicing      ☐ split()      ☒ input()

**Process (as a concise and precise LIST OF STEPS / pseudocode):**

(Assume libraries have already been imported.)

**Answer Key:**

- (a) Ask the user for input file name
  - (b) Load the data into a numpy array, call it **grid**
  - (c) Ask the user for number to search for and store it in n
  - (d) Set variables row and col to -1
  - (e) Use a nested loop to consider every number in **grid** looping for rows in outer loop and columns in inner loop
    - i. if the current number (the number at `grid[current_row, current_column]`) == n, set row to current\_row and set col to current\_column
  - (f) Return row and col. If row and col are -1, it indicates the number was not found in the grid
6. Consider the `open_restaurants.csv` dataset for **restaurant reopening applications** under Phase Two of the New York Forward Plan to place outdoor seating in front of their business on the sidewalk and/or roadway. **Each row in the dataset corresponds to an application.** A snapshot of the data is given in the image below:

| Seating Interest | Restaurant Name           | Borough   | Sidewalk Area | Roadway Area | Approved for Sidewalk Seating | Approved for Roadway Seating |
|------------------|---------------------------|-----------|---------------|--------------|-------------------------------|------------------------------|
| sidewalk         | HUNGRY GHOST              | Manhattan | 200           | 640          | yes                           | no                           |
| both             | Prince Laban&Chinese rest | Queens    | 144           | 144          | yes                           | yes                          |
| sidewalk         | Philly Pretzel Factory    | Brooklyn  | 6500          | 920          | yes                           | no                           |
| both             | BICKLES TO GO             | Bronx     | 100           | 160          | yes                           | yes                          |
| roadway          | STARBUCKS                 | Manhattan | 160           | 160          | no                            | yes                          |
| roadway          | OVENLY                    | Brooklyn  | 40            | 168          | no                            | yes                          |
| sidewalk         | LE PAIN QUOTIDIEN         | Manhattan | 105           | 280          | yes                           | no                           |
| both             | Le Pain Quotidien GCW     | Manhattan | 90            | 240          | yes                           | yes                          |
| both             | Asian Kabab and Curry     | Brooklyn  | 60            | 60           | yes                           | yes                          |

Fill in the Python program below:

### Answer Key:

```
#Import the libraries for data frames
import pandas as pd

#Prompt user for input file name:
csvFile = input('Enter CSV file name: ')

#Read input data into data frame:
df = pd.read_csv(csvFile)

#Print the 5 restaurant names with most applications
#Each row is an application, count the number of rows
per Restaurant Name and print the top 5
print(df['Restaurant Name'].value_counts()[:5])

#Group the data by Borough to extract applications in Brooklyn
#use groupby and get_group
brooklyn = df.groupby('Borough').get_group('Brooklyn')

#Print the average sidewalk area in Brooklyn
print(brooklyn['Sidewalk Area'].mean())
```

7. Consider the Python program below to display the first  $n$  squares. **Fill-in the functions** based on the comments and the overall program. Pay attention to the sample output in the comments in-order to implement the function correctly.

### Answer Key:

```
def print_n_squares(n):
 for i in range(1,n+1):
 print(i, '**2 =', i**2)
```

**Answer Key:**

```
def validate_input(num):
 while(num < 1):
 print("Please enter a positive number.")
 num = int(input("How many squared numbers to display? "))
 return num

Display numbers squared
def main():
 i = int(input("How many squared numbers to display? "))
 i = validate(i)

 #print first n squares
 print_n_squares(i)
```

8. (a) What does the MIPS program below print:

**Answer Key:**

Hello!

- (b) Modify the program to print out Hell!  
Shade in the box for each line or line-pair that needs to be changed and rewrite the instruction below. If the line needs to be deleted, write **Delete**.

**Answer Key:**

```
Print Hell!
ADDI $sp, $sp, -6
ADDI $t0, $zero, 72 # H
SB $t0, 0($sp)
ADDI $t0, $zero, 101 # e
SB $t0, 1($sp)
ADDI $t0, $zero, 108 # l
SB $t0, 2($sp)
ADDI $t0, $zero, 108 # l
SB $t0, 3($sp)
ADDI $t0, $zero, 33 # !
SB $t0, 4($sp)
```



```
ADDI $t0, $zero, 0 # (null)
SB $t0, 5($sp)
```

```
ADDI $v0, $zero, 4 # 4 is for print string
ADDI $a0, $sp, 0
syscall # print to the log
```

- (c) Modify the MIPS program below to count from 20 to 5, down by 5. Shade in the box for each line that needs to be changed and rewrite the instruction below.

**Answer Key:**

```
ADDI $s0, $zero, 20 #set s0 to 20
ADDI $s1, $zero, 5 #set s1 to 5
ADDI $s2, $zero, 5 #use to compare for branching
AGAIN: SUB $s0, $s0, $s1
BEQ $s0, $s2, DONE
J AGAIN
DONE: #To break out of the loop
```

- (d) After the modification, how many times is the line labeled **AGAIN:** executed?

**Answer Key:**

3 times.

9. Fill in the C++ programs below to produce the Output on the right.

```
#include <iostream>
using namespace std;
int main()
{
 for(int i = 3; i <=15;){
```

- (a) **Answer Key:**

```
 i += 3

 cout << i+2 << endl;
}
return 0;
}
```

```
#include <iostream>
using namespace std;
int main()
{
 int count = 20;
 int num = 10;
```

(b)

```
 while(count >=0 && num) {
 cout << count << " " << num << endl;
 count -= 2;
 num -= 5;
 }
 return 0;
}
```

**Answer Key:**

```
num >= 1
or
num > 0
#include <iostream>
using namespace std;
int main(){
 for (int i = 10; ; i--){
```

(c) **Answer Key:**

```
 i > 5
 or
 i >= 6

 cout << "Keep going!" << endl;
}
return 0;
}
```

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

```
for i in range(2,13,2):
 for j in range(10,i,-3):
 print(i, j)
```

**Answer Key:**

```
#include <iostream>
```

```
using namespace std;
int main(){
 for(int i = 2; i < 13; i+=2){
 for(int j = 10; j > i; j-=3){
 cout << i << " " << j << endl;
 }
 }
 return 0;
}
```

- (b) Write a **complete C++ program** that asks the user for an amount and outputs the interest as follows:

- "5%" if the amount is less than \$500
- "8%" if the amount is in range [\$500 , \$2000]
- "10%" otherwise

**Answer Key:**

```
//include library and namespace
#include <iostream>
using namespace std;

//function signature
int main(){
 //declare variables
 float amount;

 //obtain input
 cout << "Please enter a dollar amount: ";
 cin >> amount;

 //output interest
 if(amount < 500)
 cout << "5%" << endl;
 else if(amount <= 2000)
 cout << "8%" << endl;
 else
 cout << "10%" << endl;
 return 0;
}
```