Answer Key:

Final Exam, Version 2
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 = ' John Keats:  "Impossible is for the unwilling." '
```

i.
```
print(quote[          ])
```

Answer Key:
```
1:11
```

ii.
```
print(quote[-12:-3].                    )
```

Answer Key:
```
upper()
```

iii.
```
print("This quote has", end=" ")
print(quote.count(          )-1, 'letter o')
```

Answer Key:
```
count('o')
```

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

i.
```
letters = "A-B-C-D"
letter_list = letters.                    
```

Answer Key:
```
split('-')
```

ii.
```
for l in letter_list:
    print(                    )
```

Answer Key:

```
l.lower()
```

(c) Consider the following shell commands:

```
$ ls
bronx.csv    data    hello.py   nyc.csv   p55.cpp
```

i. What is the output for:

```
$ mv *.csv data
$ ls
```

Answer Key:

```
data hello.py p55.cpp
```

ii. What is the output for:

```
$ mkdir code
$ mkdir code/c++
$ mv hello.py p55.cpp code
$ ls
```

Answer Key:

```
code data
```

iii. What is the output for:

```
$ cd code/c++
$ mv ../p55.cpp c++
$ cd ../
$ ls
```

Answer Key:

```
c++ hello.py
```

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

Answer Key:

i. `rgb = (100, 100, 100)`

☐ black ☐ red ☐ white **X** gray ☐ purple

ii. `rgb = "#FFFFFF"`

☐ black ☐ red **X** white ☐ gray ☐ purple

iii. `rgb = (1.0, 0.0, 9.0)`
☐ black **X** red ☐ white ☐ gray ☐ purple

iv. Select the LARGEST Hexadecimal number:
☐ 0F ☐ 99 ☐ A0 **X** FF ☐ C3

v. What is the Binary number equivalent to decimal 29?
☐ 110100 **X** 011101 ☐ 101000 ☐ 000111 ☐ 101010

(b) Given the list `colors` below, fill in the code to produce the Output on the right:

```
colors = [ "red", "blue", "yellow", "orange", "green"]
```

i. **Answer Key:**
```
for i in range( 4 ):
print(words[i], end=" ")
```

Output:
```
red blue yellow orange
```

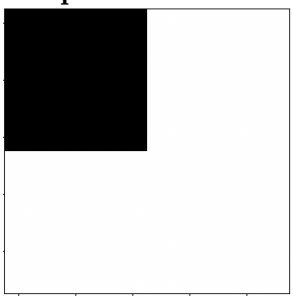
ii. **Answer Key:**
```
for j in range( 0, 5, 2 ):
  print(words[j], end=" ")
```

Output:
```
red yellow green
```

Answer Key:

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

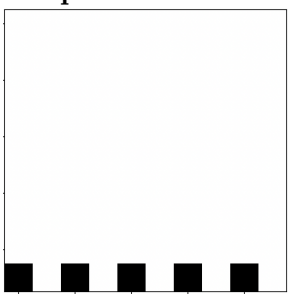
Output:



Answer Key:

iv.
```
import numpy as np
import matplotlib.pyplot as plt
im = np.ones( (10,10,3) )
im[ 9 : , : : 2 , :]  = 0
plt.imshow(im)
plt.show()
```

Output:



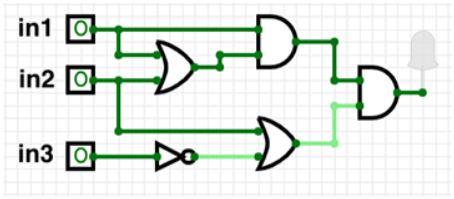
3. (a) What is the value (True/False):

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

Answer Key:
```
out = False
```

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

Answer Key:
```
out = False
```

iii.


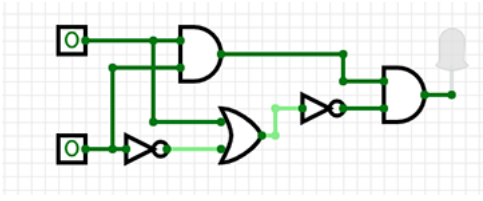

```
in1 = False
in2 = True
in3 = True
```

Answer Key:
```
out = False
```

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

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

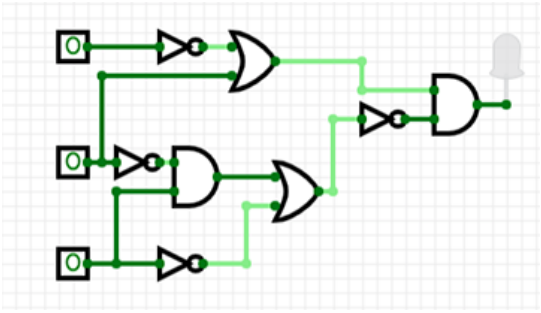
Answer Key:



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

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

Answer Key:



4. Consider the following functions:

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

```
def compare(n, c):
   return n < c


def main():
  numbers = [21, 34, 69, 62, 46, 15]
  print(count_smaller(numbers, 50))
```

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

Answer Key: n, c

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

Answer Key: numbers, 50

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

Answer Key: 6

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

Output:

Answer Key:

4

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), then outputs the index `(row, col)` of the SMALLEST number in the array.

Libraries:

Answer Key: numpy
Input:

Answer Key: The input file
Output:

Answer Key: The index of the smallest number
Design Pattern:

Answer Key: ☐ Search **X** Find Min ☐ Find Max ☐ Find All
Principal Mechanisms (select all that apply):

Answer Key: ☐ Search ☐ Single Loop **X** Nested Loop **X** Conditional (if/else) statement
XIndexing / Slicing ☐ `split()` **X** `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) Set variables min_row and min_col to 0
(d) Use a nested loop to consider every number in the `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] ¡ grid[min_row, min_col], set min_row to current_row and set min_col to current_column
(e) Return min_row and min_col

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 number of applications for each Borough
# (i.e. number of applications in Queens, number of applications in Bronx, etc.)
print(df['Borough'].value_counts())

#Group the data by Approved Sidewalk Seating to extract only those approved
#Use groupby and get_group
approved = df.groupby('Approved for Sidewalk Seating').get_group('yes')

#Print the smallest sidewalk area among the approved applications
print(approved['Sidewalk Area'].min())
```

7. Consider the Python program below to display the first 5 powers of an input number `n`. **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_5_powers(num):
  for i in range(1,6):
    print(num, '**', i, '=', num**i)
```

Answer Key:

```
def validate(n):
  while(n < 1):
     print("Please enter a positive number.")
     n = int(input("Compute first 5 powers of? "))
  return n

# Display first 5 powers of input integer
def main():
  in_num = int(input("Display first 5 powers of? "))
  in_num = validate(in_num)

  #print first 5 powers
  print_5_powers(in_num)
```

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

Answer Key:

`Hello!`

(b) Modify the program to print out `Hill`
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 Hill
ADDI $sp, $sp, -5
ADDI $t0, $zero, 72 # H
SB $t0, 0($sp)
ADDI $t0, $zero, 105 # i
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, 0 # (null)
SB $t0, 4($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 60 to 20, down by 10. Shade in the box for each line that needs to be changed and rewrite the instruction below.

Answer Key:

```
ADDI $s0, $zero, 60 #set s0 to 60
ADDI $s1, $zero, 10 #set s1 to 10
ADDI $s2, $zero, 20 #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:

4 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 = 30; i >= 5;          ){
```

(a) **Answer Key:**

```
i -= 5


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

```
#include <iostream>
using namespace std;
int main()
{
    int count = 0;
    int num = 1;
```

(b)

```
    while(count <= 20 && num [          ]){
        cout << count << " " << num << endl;
        count += 2;
        num += 5;
    }
    return 0;
}
```

Answer Key:

```
 num <= 10
or
num < 11
```

```
#include <iostream>
using namespace std;
int main(){

    for (int i = 5; [          ] ; i++){
```

(c) **Answer Key:**

```
i < 15
  or
i <= 14
```

```
          cout << "Still counting!" << endl;
    }
    return 0;
}
```

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

```
for i in range(0,9,3):
  for j in range(1,i,2):
    print(i, j)
```

Answer Key:

```
#include <iostream>
```

```
using namespace std;
int main(){
    for(int i = 0; i < 9; i+=3){
        for(int j = 1; j < i; j+=2){
            cout << i << ' ' << j << endl;
        }
    }
    return 0;
}
```

(b) Write a **complete C++ program** that asks the user for their child's age and outputs the age category on a new line as follows:

- `"Toddler"` if the child is 2 or younger
- `"Preschooler"` if the child is older than 2 but younger than 5
- `"Kid"` if the child is 5 or older and younger than 14
- `"Teen"` otherwise

Answer Key:

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

//function signature
int main(){

    //declare variables
    int age;

    //obtain input
    cout << "Please enter your child's age: ";
    cin >> age;

    //output age category
    if(age <= 2)
        cout << "Toddler" << endl;
    else if(age < 5)
        cout << "Preescholer" << endl;
    else if(age < 14)
        cout << "Kid" << endl;
    else
        cout << "Teen" << endl;
    return 0;
}
```