Answer Key:

Final Exam, Version 4
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 = ' "Simplicity is the ultimate sophistication." Leonardo da Vinci'
```

i. `print(` ☐ `)`

Answer Key:
`quote[-17:-9]`

ii. `print(quote` ☐ `)`

Answer Key:
`[2:12].lower()`

iii. `words =` ☐
`print("This quote has", len(words)-4, "words")`

Answer Key:
`quote.split(" ")`

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

```
letters = "z * y * x * w"
```

i. `print("There are", letters.` ☐ `, "letters")`

Answer Key:
`count('*')+1`

ii.
```
for i in range(len(letters)):
    if ☐ :
```

```
            print(letters[i])
```

Answer Key:

```
i % 4 == 0
```

(c) Consider the following shell commands:

```
$ ls
code web
```

i. What is the output for:

```
$ cd code
$ ls
plots   star.py    turtle_progs
$ mv star.py turtle_progs/
$ ls
```

Answer Key:

```
plots turtle_progs
```

ii. What is the output for:

```
$ cd turtle_progs/
$ ls
panorama.py  ramble.py  star.py
$ ls | grep ra*
```

Answer Key:

```
ramble.py
```

iii. What is the output for:

```
$ cd ../ ../
$ ls
```

Answer Key:

```
code    web
```

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

Answer Key:

i. `rgb = (255, 0, 0)`
☐ black **X** red ☐ white ☐ gray ☐ purple

ii. `rgb = "#ABABAB"`
☐ black ☐ red ☐ white **X** gray ☐ purple

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

iv. Select the LARGEST Binary number:
X 110100 ☐ 011101 ☐ 101000 ☐ 000111 ☐ 101010

v. What is the Binary number equivalent to decimal 160?
☐ 0F ☐ 99 **X** A0 ☐ FF ☐ C3

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

```
symbols = [ "*", "#", "+", "$"]
```

i. **Answer Key:**

```
for i in range( 4 ):
print(symbols[i], end=" ")
```

Output:

```
* # + $
```

ii. **Answer Key:**

```
for j in range( 0, 4, 3 ):
  print(symbols[j], end=" ")
```

Output:

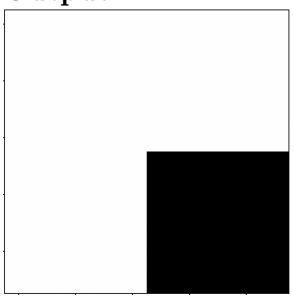
```
* $
```

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:

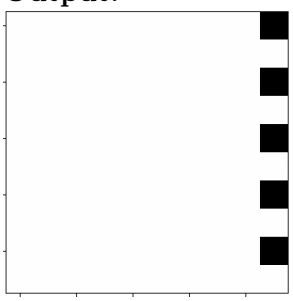


Answer Key:

iv.

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

Output:



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

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

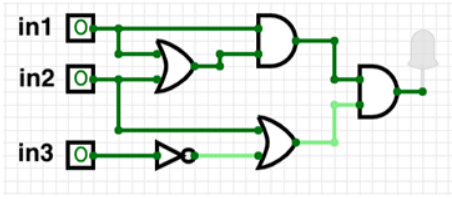
Answer Key:

```
out = False
```

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

Answer Key:

```
out = True
```

iii.



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

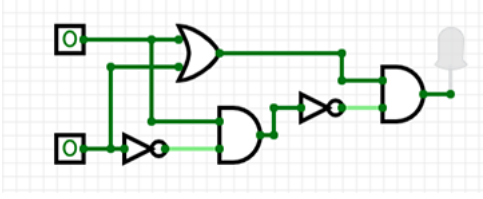
Answer Key:

```
out = True
```

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

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

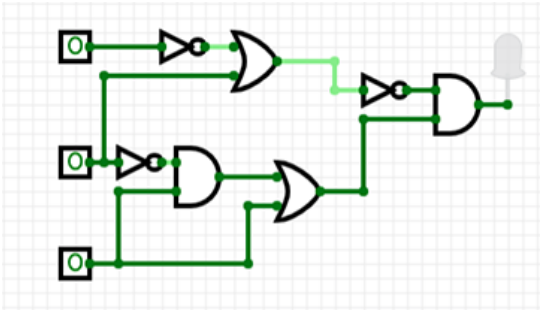
Answer Key:



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

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

Answer Key:



4. Consider the following functions:

```
def add_odd(items):
  sum = 0
  for i in range(len(items)):
    sum += compare(items[i])
  return sum
```

```
def compare(i):
  return i * (i % 2)


def main():
  nums = [1, 2, 3, 4, 5, 6, 7, 8, 9]
  print(add_odd(nums))
```

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

   Answer Key: i

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

   Answer Key: nums

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

   Answer Key: 9

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

   Output:

   Answer Key:
   25

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 number of occurrences of `n` found in the grid.
   Libraries:

Answer Key: numpy
Input:

Answer Key: The input file and number n
Output:

Answer Key: The number of times n is found in the grid
Design Pattern:

Answer Key: ☐ Search ☐ Find Min ☐ Find Max **X** 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) Ask the user for input number and store it in n
(d) Set variables count to 0
(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, increment count
(f) Return count

6. Consider the `class_size.csv` dataset from NYC Open Data **preliminary average class size aggregated by school for 2021**. Each row in the dataset corresponds to a class grade level and program type at a given school. A snapshot of the data is given in the image below:

School Name	Grade Level	Program Type	Num Students	Num Classes	Avg Class Size	Min Class Size	Max Class Size
BROOKLYN ARBOR	K	Gen Ed	41	2	20.5	19	22
BROOKLYN ARBOR	K	ICT	19	1	19.0	19	19
BROOKLYN ARBOR	1	Gen Ed	60	3	20.0	18	22
BROOKLYN ARBOR	1	ICT	16	1	16.0	16	16
BROOKLYN ARBOR	2	Gen Ed	48	2	24.0	23	25
BROOKLYN ARBOR	2	ICT	44	2	22.0	21	23
BROOKLYN ARBOR	3	Gen Ed	70	3	23.3	21	25
BROOKLYN ARBOR	3	ICT	26	1	26.0	26	26
BROOKLYN ARBOR	4	Gen Ed	42	2	21.0	19	23
BROOKLYN ARBOR	4	ICT	48	2	24.0	23	25

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 rows per Program Type
# (i.e. number of rows for Gen Ed, number of rows for ICT, etc.)
print(df['Program Type'].value_counts())

#Group the data by Grade Level to extract Kindergarten
#use groupby and get_group
kindergarten = df.groupby('Grade Level').get_group('K')

#Print the average class size for kindergarten across all schools
print(kindergarten['Avg Class Size'].mean())
```

7. Consider the Python program below to display the first n Fibonacci numbers. The Fibonacci sequence is generated as follows: F0 = 0, F1 = 1, F2 = F1 + F0, F3 = F2 + F1, ... , Fn = Fn-1 + Fn-2. **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_fib(n):
  f_1 = 0
```

```
  f_2 = 1
  print('F0 = 0')
  for i in range(1,n+1):
    fib = f_1 + f_2
    print('F'+str(i), '=', fib)
    f_2 = f_1
    f_1 = fib
```

Answer Key:

```
def validate_input(num):
  while(num <= 2):
     print("Please enter a number > 2.")
     num = int(input("How many Fibonacci numbers to print? "))
  return num

# Display n Fibonacci numbers
def main():
  n = int(input("How many Fibonacci numbers to print? "))
  n = validate(n)

  #print n Fibonacci numbers
  print_n_fib(n)
```

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

Answer Key:

```
Hello!
```

(b) Modify the program to print out `HELLO`
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 HELLO
ADDI $sp, $sp, -6
ADDI $t0, $zero, 72 # H
SB $t0, 0($sp)
ADDI $t0, $zero, 69 # E
SB $t0, 1($sp)
ADDI $t0, $zero, 76 # L
SB $t0, 2($sp)
ADDI $t0, $zero, 76 # L
```

```
SB $t0, 3($sp)
ADDI $t0, $zero, 79 # 0
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 10 to 30, up by 5. Shade in the box for each line that needs to be changed and rewrite the instruction below.

Answer Key:

```
ADDI $s0, $zero, 10 #set s0 to 10
ADDI $s1, $zero, 5 #set s1 to 5
ADDI $s2, $zero, 30 #use to compare for branching
AGAIN: ADD $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 = 0; [          ] i += 10){
```

(a) **Answer Key:**

```
    i <=30;


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

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

(b)

```
    while(count [      ] && num [      ]){
        cout << count << " " << num << endl;
        count +=1;
        if(count % 2 == 0)
            num +=1;
    }
    return 0;
}
```

Answer Key:

```
count <= 5 \&\& num <= 2
or
count < 6 \&\& num <3
```

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

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

(c) **Answer Key:**

```
  i >= -2;
  or
  i > -3;

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

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

```
for i in range(20,3,-5):
  for j in range(50,i,-3):
    print(i, j)
```

Answer Key:

```
#include <iostream>
using namespace std;
int main(){
   for(int i = 20; i > 3; i-=5){
        for(int j = 50; j > i; j-=3){
            cout << i << " " << j << endl;
        }
    }
    return 0;
}
```

(b) Write a **complete C++ program** that asks the user for the number of credit hours and outputs the student category on a new line as follows:

- `"Freshman"` for [0,29] hours of earned credit
- `"Sophomore"` for [30,59] hours of earned credit
- `"Junior"` for [60,89] hours of earned credit
- `"Senior"` 90 or more hours of earned credit

Answer Key:

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

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

    //obtain input
    cout << "Please enter your credit hours: ";
    cin >> hours;

    //output student category
    if(hours <= 29)
        cout << "Freshman" << endl;
    else if(hours <= 59)
        cout << "Sophomore" << endl;
    else if(hours <= 89)
        cout << "Junior" << endl;
    else
        cout << "Senior" << endl;
    return 0;
}
```