Answer Key:

FINAL EXAM, VERSION 2
CSci 127: Introduction to Computer Science
Hunter College, City University of New York

23 May 2022

1. (a) Fill in the code below to produce the Output on the right:

    ```
    workdays = "Monday?Tuesday?Wednesday?Thursday?"
    summer_months = "*June*July*August*"
    long_weekend = "Friday_Saturday_Sunday"
    seasons = "+Spring+Summer+Fall+Winter"
    ```

    i.
    ```
    print(            ],             ])
    ```

    Answer Key:
    ```
    print(summer_months[1:5], workdays[-9:-1])
    ```

    ii.
    ```
    months = summer_months[          ].split(          )

    print("Summer has" , len(          ), "months.")
    ```

    Answer Key:
    ```
    months = summer_months[1:-1].split('*')
    print("Summer has", len(months), "months.")
    ```

    iii.
    ```
    for m in

        print(          )
    ```

    Answer Key:
    ```
    for m in months:
        print(m.lower())
    ```

    (b) Consider the following shell commands:

```
$ ls
hello.py   pictures     pp_hello.cpp     temp
```

   i. What is the output for:
```
$ mv hello.py    p1.py
$ ls
```

     Answer Key:
```
p1.py    pictures     pp_hello.cpp      temp
```

  ii. What is the output for:
```
$ mkdir python
$ mv *.py   python
$ ls
```

     Answer Key:
```
python   pictures pp_hello.cpp temp
```

iii. What is the output for:
```
$ cd python
$ mkdir p50_60
$ mkdir py_5
$ ls | grep py
```

     Answer Key:
```
p1.py    py_5
```

2. (a) Select the correct option.

   Answer Key:

   i. What color is tina? `tina.color(0.5,0.5,0.5)`
     ☐ black   ☐ red   ☐ white   **X** gray   ☐ purple

  ii. Select the SMALLEST Binary number:
     ☐ 0110   ☐ 1001   ☐ 1101   ☐ 1011   **X** 0000

iii. Select the LARGEST Hexadecimal number:
     ☐ 0A   ☐ 22   ☐ A0   **X** FF   ☐ CD

iv. What is the decimal number equivalent to binary 10110?
     ☐ 16   ☐ 25   ☐ 24   **X** 22   ☐ 18

  v. What is the decimal number equivalent to hexadecimal 18?

X 24   ☐ 19   ☐ 28   ☐ 13   ☐ 23

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

```
nums = [ 23, 45, 76, 23, 98, 45 , 11, 4, 33, 29, 5, 66]
```

i. **Answer Key:**
```
for i in range( 5,11 ):
print(nums[i], end=" ")
```

Output:

45 11 4 33 29 5

ii. **Answer Key:**
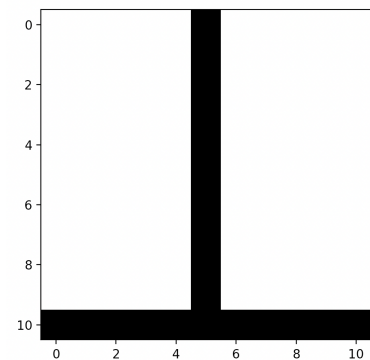```
for j in range(0,11,3):
    print(nums[j], end=" ")
```

**Output:**

| 23 23 11 29 |
|---|

**Answer Key:**

```
import numpy as np
import matplotlib.pyplot as plt
img = np.ones((11,11,3))
```
iii.

img[ **-1** , **:** , :] = 0

img[ **:** , **5** , :] = 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) or (not in1 and in2)
```

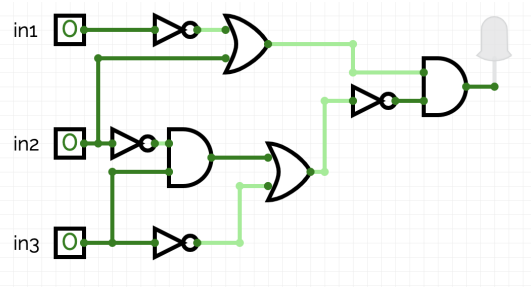**Answer Key:**
```
out = False
```

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

Answer Key:
```
out = True
```
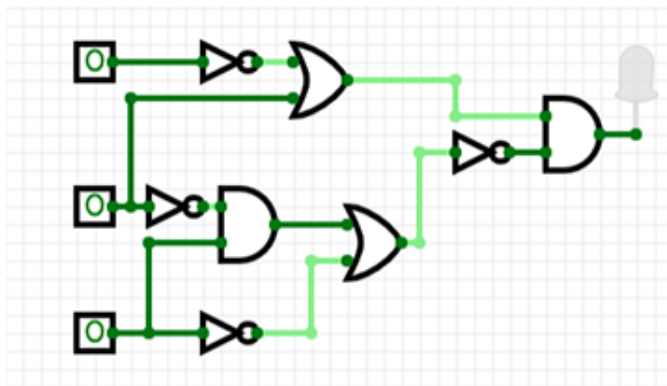
iii.

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

Answer Key:

```
out = False
```

(b) Draw a circuit 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 meow(n, ch):
    for i in range(1,n):
        woof(i, ch)
        print()
```

```
def woof(i, l):
    for j in range(i):
        print(l, end='')


def main():
    meow(6, 'X')
```

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

Answer Key: n, ch

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

Answer Key: i, ch

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

Answer Key: 5

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

Output:

Answer Key:

```
X
XX
XXX
XXXX
XXXXX
```

5. Design an algorithm that asks the user for the name of a csv file and the name of a column, then returns the number values in that column that repeat more than once. For example, if the column contains values [a, b, b, a, c, c, c, d, e, f], the program returns 3, because a, b and c are repeated, while d, e and f occur only once. You must write detailed **pseudocode** as a precise list of steps that completely and precisely describe the algorithm.

Libraries (if any):

Answer Key: pandas
Input:

Answer Key: The file name and the column name
Output:

Answer Key: An integer, the number of values that repeat more than once in the column

Principal Mechanisms (select all that apply):

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

X Indexing / Slicing   ☐ `split()`   **X** `input()`

Process (as a concise and precise LIST OF STEPS / pseudocode):
(Assume libraries, if any, have already been imported.)

Answer Key:

(a) Ask the user for the name of a csv file

(b) Ask the user for the name of a column in that csv file

(c) Read the csv file into a dataframe

(d) Use `value_counts()` to count the number of occurrences for each value in that column and assign it to a variable, say `count_list`

(e) start a counter at 0

(f) Loop for every value in `count_list` and increment the counter if that value is `>1`

(g) Output the count

(h) ALTERNATIVELY: Output: `len(count_list[count_list > 1])`

6. Consider `apple_stocks.csv` from the Apple Stock Price dataset from kaggle, reporting Apple's stock prices (in USD $) from December 1980 to May 2022. **Each row in the dataset corresponds to the stock values for one day of trading**. A snapshot of the data is given in the image below:

Date	Open	High	Low	Close	Volume
1980-12-12	0.128348	0.128906	0.128348	0.128348	469033600
1980-12-15	0.122210	0.122210	0.121652	0.121652	175884800
1980-12-16	0.113281	0.113281	0.112723	0.112723	105728000
1980-12-17	0.115513	0.116071	0.115513	0.115513	86441600
■ ■ ■					
2022-04-28	159.25000	164.52000	158.92999	163.63999	130216800
2022-04-29	161.83999	166.19999	157.25000	157.64999	131587100
2022-05-02	156.71000	158.22999	153.27000	157.96000	123055300

Fill in the Python program below:

Answer Key:

```
#Import the libraries for plotting and data frames
import pandas as pd
import matplotlib.pyplot as plt

#Prompt user for input file name:
in_file = input("Please enter the name of the Apple stocks csv file: ")
```

```python
#Read input data into data frame:
apple = pd.read_csv(in_file)

#Print the highest opening value
print(apple["Open"].max())

#Print the average closing value
print(apple["Close"].mean())

#Print the difference between the last (2022-05-02) and first (1980-12-12) High values
print(apple["High"][len(apple)-1] - apple["High"][0])

#Plot the closing values against the date
apple.plot(x="Date", y="Close")
plt.show()
```

7. Fill in the following functions that are part of a program that maps GIS data :

- **getData()**: asks the user for latitude and longitude of the user's current location and returns those as floating points numbers
- **mark()**: creates and returns a folium marker at coordinates lat, lon
- **saveMap()**: adds mark to a map and saves it with name "my_map.html"

Answer Key:

```python
import folium
def getData():
    """
    Asks the user for latitude and longitude of the user's current location
    Returns lat, lon as floating points numbers
    """
    lat = float(input('Enter current latitude: '))
    lon = float(input('Enter current longitude: '))
    return(lat, lon)

def mark(lat, lon):
    """
    Creates and returns a folium marker at coordinates lat, lon
    """
    return folium.Marker(location = [lat, lon])

def saveMap(mark):
    """
    Adds mark to a map and saves it with name "my_map.html"
```

```
    """
    map = folium.Map()
    mark.add_to(map)
    map.save(outfile="my_map.html")
```

8. (a) What is printed by the MIPS program below:

   Answer Key:

   BBBBBBBBBBBBBB

   (b) Modify the program to print out "ADGJ". Shade in the box for each line that needs to be changed and rewrite the instruction below, or add instructions where necessary.

   Answer Key:

```
#Loop through characters
ADDI $sp, $sp, -5      # Set up stack
ADDI $s3, $zero, 1      # Store 1 in a register
ADDI $t0, $zero, 65     # Set $t0 at 65 (A)
ADDI $s2, $zero, 5      # Use to test when you reach 5
SETUP: SB $t0, 0($sp)    # Next letter in $t0
ADDI $sp, $sp, 1         # Increment the stack
ADDI $s3, $s3, 1         # Increment the counter by 1
ADDI $t0, $t0, 3         # Increment the letter (added instruction)
BEQ $s3, $s2, DONE       # Jump to done if $s3 == 5
J SETUP                  # If not, jump back to SETUP for loop
DONE: ADDI $t0, $zero, 0 # Null (0) to terminate string
SB $t0, 0($sp)           # Add null to stack
ADDI $sp, $sp, -4      # Set up stack to print
ADDI $v0, $zero, 4       # 4 is for print string
ADDI $a0, $sp, 0         # Set $a0 to stack pointer for printing
syscall                  # Print to the log
```

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

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

(a)
```
    for(          ; i <=35;          ){

        cout << i+3  << endl;
    }
    return 0;
}
```

Answer Key:

```
for( int i = 2; i <=35; i +=5)
#include <iostream>
using namespace std;
int main()
{
    int n=-14, m=10;
```

(b)
```
    while(n+m          ){

        cout << n << " " << m << endl;
        n+=2;
        m--;
    }
    return 0;
}
```

Answer Key:

```
while(n+m < -1)
```

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

for ( [                    ] ){
```

Answer Key:

```
for(int i = 2; i < 8; i++)
```

(c)   for([   ]){

Answer Key:

```
for(int j = 10; j >= i; j--)

                cout << i << j-i << " ";
        }
        cout << endl;
    }
    return 0;
}
```

10. (a) Write a **complete C++ program** that repeatedly asks the user for two amounts until their sum is positive, then it outputs the sum:

Answer Key:

```
#include <iostream>
using namespace std;

int main()
{
    float amount1=0.0, amount2=0.0;
    do{
        cout << "Please enter an amount: ";
        cin >> amount1;
        cout << "Please enter another amount: ";
        cin >> amount2;

    }while(amount1+amount2<1);

    cout << "The sum is " << amount1+amount2 << endl;
    return 0;
}
```

(b) Write a **complete C++ program** that asks the user for an amount and computes the

number of years it takes to double the amount, if it is subject to an increase of 25% each year.

Answer Key:

```cpp
#include <iostream>
using namespace std;

int main()
{
    float amount = 0.0;
    int year = 0;

    cout << "Please enter an mount: ";
    cin >> amount;
    int doubled_amount = amount*2;
    while(amount < doubled_amount){
        amount +=  amount*0.25;
        year +=1;
        cout << year << " " << amount << endl;
    }

    cout << "It took " << year << " years to double your amount to " << amount << endl
    return 0;
}
```

SCRATCH PAPER (page left intentionally blank)

SCRATCH PAPER (page left intentionally blank)