Row:	Seat:

Final Exam, Version 1
CSci 127: Introduction to Computer Science
Hunter College, City University of New York

May 17, 2023

Exam Rules

- Show all your work. Your grade will be based on the work shown.
- The exam is closed book and closed notes with the exception of an 8 1/2” x 11” piece of paper filled with notes, programs, etc.
- When taking the exam, you may have with you pens and pencils, and your note sheet.
- You may not use a computer, calculator, tablet, phone, earbuds, or other electronic device.
- **Do not open this exam until instructed to do so.**

Hunter College regards acts of academic dishonesty (e.g., plagiarism, cheating on examinations, obtaining unfair advantage, and falsification of records and official documents) as serious offenses against the values of intellectual honesty. The College is committed to enforcing the CUNY Policy on Academic Integrity and will pursue cases of academic dishonesty according to the Hunter College Academic Integrity Procedures.

I understand that all cases of academic dishonesty will be reported to the Dean of Students and will result in sanctions.
Name:
EMPLID: □□□□□□□□
Email:
Signature:

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

```
workdays = "Monday=Tuesday=Wednesday=Thursday=Friday"
winter = "^^December^^January^^February^^"
weekend = "Saturday*Sunday"
classes = "(Math(Science(English(History"
```

i.

```
print( workdays[-15:-7] , winter[2:10] )
```

Output:

```
Thursday December
```

ii.

```
four_classes = classes[ 1: ].split( "(" )

print("There are", len( four_classes ),"classes.")
```

Output:

```
There are 4 classes.
```

iii.

```
for s in four_classes:
    print( s.lower() )
```

Output:

```
math
science
english
history
```

(b) Consider the following shell commands:

```
$ pwd
/Users/guest
$ ls
bronx.png  circuit.txt   nand.txt  nyc.png    hello
```

i. What is the output for:

```
$ mkdir data
$ mv *txt data
$ ls
```

Output:

```
bronx.png data hello nyc.png
```

ii. What is the output for:

```
$ cd data
$ ls
```

Output:

```
circuit.txt nand.txt
```

iii. What is the output for:

```
$ cd ../hello
$ pwd
```

Output:

```
/Users/guest/hello
```

2. (a) Select the correct option.

   i. What color is tina after this command? `tina.color(0.0,1.0,0.0)`
   ☐ black ☐ red ☐ white ☐ gray ☒ green

   ii. Select the SMALLEST Binary number:
   ☐ 1011 ☐ 1101 ☒ 0111 ☐ 1010 ☐ 1110

   iii. Select the LARGEST Hexadecimal number:
   ☐ AA ☐ BA ☒ DC ☐ CC ☐ CD

   iv. What is the binary number equivalent to decimal 11?
   ☒ 1011 ☐ 1101 ☐ 0111 ☐ 1010 ☐ 1110

   v. What is the hexadecimal number equivalent to decimal 166?
   ☒ A6 ☐ AA ☐ FC ☐ DC ☐ CD

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

```
nums = [ 33, 44, 214, 54, 765, 4321, 34, 23]
```

   i.

```
for i in range( 2 , 4 ):
    print(nums[i], end=" ")
```

Output:

```
214 54
```

   ii.

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

Output:

```
44 54 4321
```

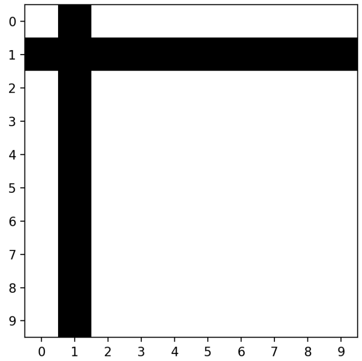
   iii.

```
import numpy as np
import matplotlib.pyplot as plt
img = np.ones( (10,10,3) )

img[ : , 1 , :]  = 0 # black column

img[ 1 , : , :]  = 0 # black row
plt.imshow(img)
plt.show()
```

Output:



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

i.

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

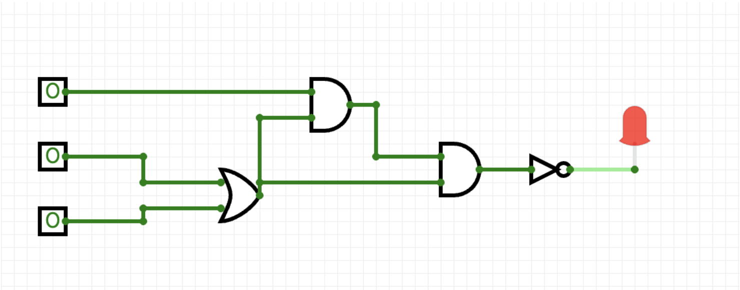
□ True ☒ False

ii.

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

□ True ☒ False

iii.

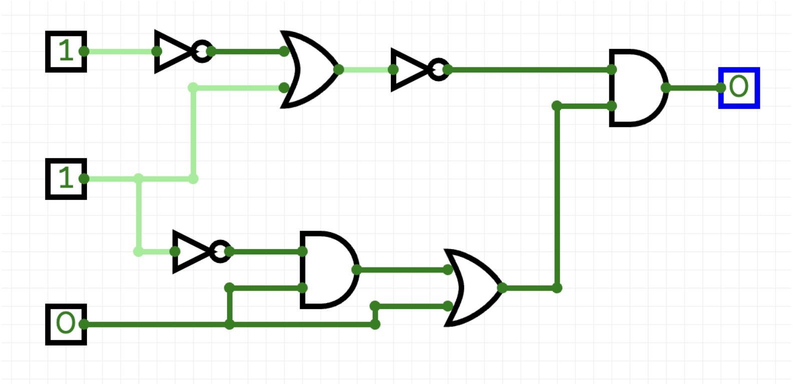


```
in1 = True
in2 = False
in3 = True
```

□ True ☒ False

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

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



4. Consider the following functions:

```
def hello(x, y):
 for i in range(x):
   if(i % 3 == 0):
    print(world(i, y))
```

```
def world(i, z):
  for j in range(i):
    z+=3
  return z


def main():
   hello(4, 12)
```

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

x,y

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

i,y

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

2

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

Output:

```
12
21
```

5. Design an algorithm that first asks the user for a name of an image .png file and the name of an output file. Your algorithm should then create a new image that has only the green and blue channels of the original image. You must write detailed **pseudocode** as a precise list of steps that completely describes the algorithm.

Libraries (if any): `matplotlib.pyplot`

Input: `names of input and output image files`

Output: `image with only the green and blue color channels of the original`

Principal Mechanisms (select all that apply):

□ Single Loop □ Nested Loop □ Conditional (if/else) statement

☒ Indexing / Slicing □ `split()` ☒ `input()`

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

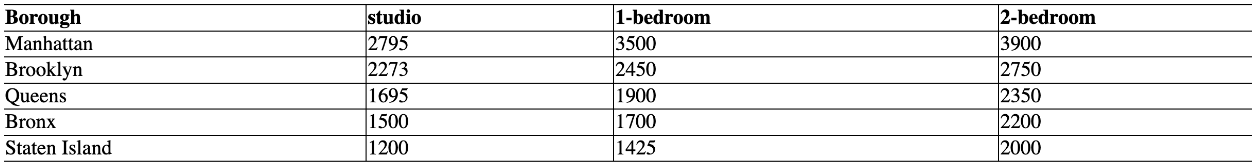
(a) Ask user for the name of the input and output image files

(b) Read the input image file into a numpy array using `plt.imread()`

(c) Set the red channel of the image array to 0 (`img[:,:,0] = 0`)

(d) Save the modified array as a new image using the output file name given

6. Consider the following data which shows the average rent price based on the number of rooms the apartment has. Each row in the data represents the average prices for the different boroughs. A snapshot is given in the image below:



Borough	studio	1-bedroom	2-bedroom
Manhattan	2795	3500	3900
Brooklyn	2273	2450	2750
Queens	1695	1900	2350
Bronx	1500	1700	2200
Staten Island	1200	1425	2000

Fill in the Python program below:

```
#Import the libraries for data frames
```

```
import pandas as pd
```

```
#Prompt user for input file name:
```

```
csvFile = input("Enter the file:  ")
```

```
#Read input data into data frame:
```

```
df = pd.read_csv(csvFile)
```

```
#Create a new column in the dataframe that represents the overall average
# apartment price for each borough (i.e. the average of the studio,
# one-bedroom, and two-bedroom prices)
```

```
df["average"] = (df["studio"]+df["1-bedroom"]+df["2-bedroom"])/3
```

7. Fill in the following functions that are part of a program that draws with turtles:

   - `getData()`: asks the user for the color and shape of a turtle and the number of sides of a polygon
   - `getTurtle()`: returns a turtle with color and shape
   - `drawPolygon()`: draws a polygon with n sides using turtle t

```
import turtle
def getData():
    """
    Asks the user for the color and shape of a turtle
    and the number of sides of a polygon.
    Returns the color and shape as strings and the sides as integer.
    """
```

```
    color = input("Enter turtle color: ")
    shape = input("Enter turtle shape: ")
    numSides = input("Enter number of sides: ")
    return color, shape, int(numSides)
```

```
def getTurtle(color, shape):
    """
    Returns a turtle with color and shape
    """
```

```
    tina = turtle.Turtle()
    tina.color(color)
    tina.shape(shape)
    return tina
```

```
def drawPolygon(t, n):
    """
    Draws a polygon with n sides using turtle t

    """
```

```
  for i in range(n):
    t.forward(50)
    t.right(360/n)
```

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

Output:

abcde

(b) Modify the program to print out `"ABCD"`. Shade in the box for each line that needs to be changed and rewrite the instruction next to the line chosen.

☒ `ADDI $sp, $sp, -6` `Answer: ADDI $sp, $sp, -5`

□ `ADDI $s3, $zero, 1`

☒ `ADDI $t0, $zero, 97` `Answer: ADDI $t0, $zero, 65 #(A)`

☒ `ADDI $s2, $zero, 5` `Answer: ADDI $s2, $zero, 4`

□ `SETUP: SB $t0, 0($sp)`

□ `ADDI $sp, $sp, 1`

□ `SUB $s2, $s2, $s3`

□ `ADDI $t0, $t0, 1`

□ `BEQ $s2, $zero, DONE`

□ `J SETUP`

□ `DONE: ADDI $t0, $zero, 0`

□ `SB $t0, 0($sp) # Add null to stack`

☒ `ADDI $sp, $sp, -5` `Answer: ADDI $sp, $sp, -4`

□ `ADDI $v0, $zero, 4 # 4 is for print string`

□ `ADDI $a0, $sp, 0 # Set $a0 to stack pointer`

□ `syscall # Print to the log`

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

(a)

```
#include <iostream>
using namespace std;
int main()
{
    for( int i = 4 ; i < 15; i+=2 ){
       cout << i-3 << endl;
    }
    return 0;
}
```

Output:

```
1
3
5
7
9
11
```

(b)

```
#include <iostream>
using namespace std;
int main()
{
     int n=12, m=-5;

    while(n > 4 && m < 0){
        n-=2;
        m++;
        cout << n << " " << m << endl;
    }
    return 0;
}
```

Output:

```
10 -4
8 -3
6 -2
4 -1
```

(c)

```
#include <iostream>
using namespace std;
int main(){
for ( for(int i = 5; i > 2; i--) ){
    cout << i;
    for( for(int j = 0; j <= i; j++) ){
           cout << "^_^  ";
        }
        cout << endl;
    }
    return 0;
}
```

Output:

```
5^_^  ^_^  ^_^  ^_^  ^_^  ^_^
4^_^  ^_^  ^_^  ^_^  ^_^
3^_^  ^_^  ^_^  ^_^
```

10. (a) Write a **complete C++ program** that repeatedly asks the user for a message until the entered message is not longer than 8 characters.

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

int main() {
    string message;

    do {
        cout << "Enter a message: ";
        cin >> message;
    }
    while(message.length() > 8);

    cout << message << endl;
}
```

(b) You have a backyard pond but the population of frogs is declining every year. You know that the pond's frog population is 1,000 and you ask an expert to calculate how many frogs are lost per year. Write a **complete C++ program** that takes the expert's number in as input and calculates the number of years it will take for the frog population to go below 50.

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

int main() {
    int numFrogs = 1000;
    int numYears = 0;

    int frogsLostYearly;
    cin >> frogsLostYearly;

    while(numFrogs >= 50) {
        numFrogs = numFrogs - frogsLostYearly;
        numYears++;
    }

    cout << numYears << endl;
}
```

SCRATCH PAPER (page left intentionally blank)

SCRATCH PAPER (page left intentionally blank)