Mock Final Exam
CSCI 127: Introduction to Computer Science
Hunter College, City University of New York

May 14, 2024

Exam Rules

- Show all your work. Your grade will be based on the work shown.
- The exam is closed book and closed notes.
- When taking the exam, you may have with you pens, pencils, and an 8 1/2" x 11" piece of paper filled with notes, programs, etc.
- You may not use a computer, calculator, tablet, smart watch, 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: Melissa Lynch
EmpID:
Signature:

ASCII TABLE

Decimal	Hex	Char	Decimal	Hex	Char	Decimal	Hex	Char	Decimal	Hex	Char
0	0	*[NULL]*	32	20	*[SPACE]*	64	40	@	96	60	`
1	1	*[START OF HEADING]*	33	21	!	65	41	A	97	61	a
2	2	*[START OF TEXT]*	34	22	"	66	42	B	98	62	b
3	3	*[END OF TEXT]*	35	23	#	67	43	C	99	63	c
4	4	*[END OF TRANSMISSION]*	36	24	$	68	44	D	100	64	d
5	5	*[ENQUIRY]*	37	25	%	69	45	E	101	65	e
6	6	*[ACKNOWLEDGE]*	38	26	&	70	46	F	102	66	f
7	7	*[BELL]*	39	27	'	71	47	G	103	67	g
8	8	*[BACKSPACE]*	40	28	(	72	48	H	104	68	h
9	9	*[HORIZONTAL TAB]*	41	29	)	73	49	I	105	69	i
10	A	*[LINE FEED]*	42	2A	*	74	4A	J	106	6A	j
11	B	*[VERTICAL TAB]*	43	2B	+	75	4B	K	107	6B	k
12	C	*[FORM FEED]*	44	2C	,	76	4C	L	108	6C	l
13	D	*[CARRIAGE RETURN]*	45	2D	-	77	4D	M	109	6D	m
14	E	*[SHIFT OUT]*	46	2E	.	78	4E	N	110	6E	n
15	F	*[SHIFT IN]*	47	2F	/	79	4F	O	111	6F	o
16	10	*[DATA LINK ESCAPE]*	48	30	0	80	50	P	112	70	p
17	11	*[DEVICE CONTROL 1]*	49	31	1	81	51	Q	113	71	q
18	12	*[DEVICE CONTROL 2]*	50	32	2	82	52	R	114	72	r
19	13	*[DEVICE CONTROL 3]*	51	33	3	83	53	S	115	73	s
20	14	*[DEVICE CONTROL 4]*	52	34	4	84	54	T	116	74	t
21	15	*[NEGATIVE ACKNOWLEDGE]*	53	35	5	85	55	U	117	75	u
22	16	*[SYNCHRONOUS IDLE]*	54	36	6	86	56	V	118	76	v
23	17	*[ENG OF TRANS. BLOCK]*	55	37	7	87	57	W	119	77	w
24	18	*[CANCEL]*	56	38	8	88	58	X	120	78	x
25	19	*[END OF MEDIUM]*	57	39	9	89	59	Y	121	79	y
26	1A	*[SUBSTITUTE]*	58	3A	:	90	5A	Z	122	7A	z
27	1B	*[ESCAPE]*	59	3B	;	91	5B	[	123	7B	{
28	1C	*[FILE SEPARATOR]*	60	3C	<	92	5C	\	124	7C	|
29	1D	*[GROUP SEPARATOR]*	61	3D	=	93	5D	]	125	7D	}
30	1E	*[RECORD SEPARATOR]*	62	3E	>	94	5E	^	126	7E	~
31	1F	*[UNIT SEPARATOR]*	63	3F	?	95	5F	_	127	7F	*[DEL]*

(Image from wikipedia commons)

1. (a) What will the following Python code print:

i.
```
banana = "fghBaaabbBTRkm"
print(banana.count("b"))
```

Output: 2

ii.
```
B = banana.split("B")
print(B[0])
```

Output: fgh

iii.
```
up = B[-1].lower()
print(up)
```

Output: trkm

iv.
```
for c in up:
    print(c.upper())
```

Output:
T
R
K
M

(b) Consider the contents of the current directory:

```
banana.txt  banana.py   carrot.csv  clementine.py   dragonfruit
```

i. What is the output for:

```
$ ls *b*
```

Output:
banana.txt
banana.py

ii. What is the output for:

```
$   mv *.py ./dragonfruit
$   ls
```

Output:
banana.txt carrot.csv
dragonfruit

iii. What is the output for:

```
$ ls -l | grep "fr" | wc -l
```

Output: 1

2. Complete the Python program below:

```
#import the libraries for image processing
```

```
import matplotlib.pyplot as plt
import numpy as np
```

```
#get a number for the color channel from user input
```

```
color= int(input("Enter a color: "))
```

```
#create an all-black image with a height of 200 and a width of 100
```

```
img = np.zeros((200, 100, 3))
```

```
if color > 2:
  exit() #exits the program
#else if the color channel is less than 0, exit the program
```

```
elif color < 0:
    exit()
```

```
#else modify the image such that the left half becomes the color entered
```

```
else:
  img[:, :50, color] = 1.0
```

```
#save the image in a file called "final.png"
```

```
plt.imsave("Final.png", img)
```

3. (a) Select the correct option.

i. What color is tina after this command? `tina.color("#111111")`
☐ black ☐ red ☐ white ☒ gray ☐ green

ii. Select the LARGEST binary number:
☐ 1011 ☒ 1101 ☐ 0111 ☐ 1010 ☐ 1001

iii. Select the LARGEST hexadecimal number:
☒ FD ☐ EA ☐ EF ☐ FC ☐ CD

iv. What is the binary number equivalent to decimal 25?
☐ 01011 ☐ 10001 ☐ 11100 ☐ 10111 ☒ 11001

v. What is the hexadecimal number equivalent to decimal 48?
☐ 34 ☐ 22 ☐ 24 ☐ 3B ☒ 30

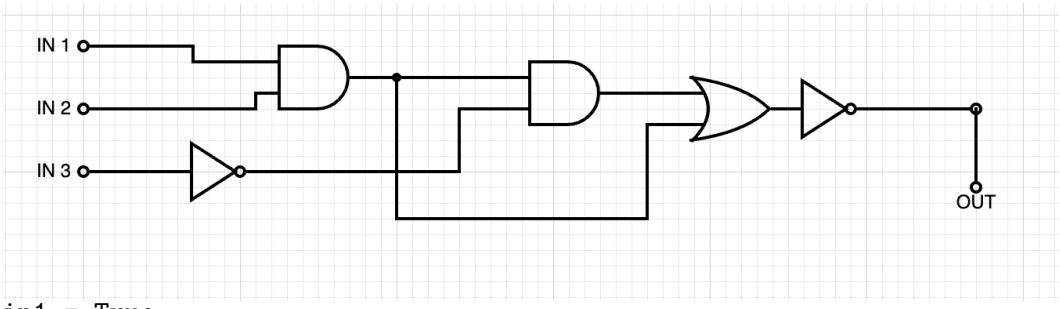
(b) i. What is the value (True/False):

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

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

C.
```
in1 = False
in2 = True or not in1
in3 = in1 and in2
out = in1 and not in3
```
out = False

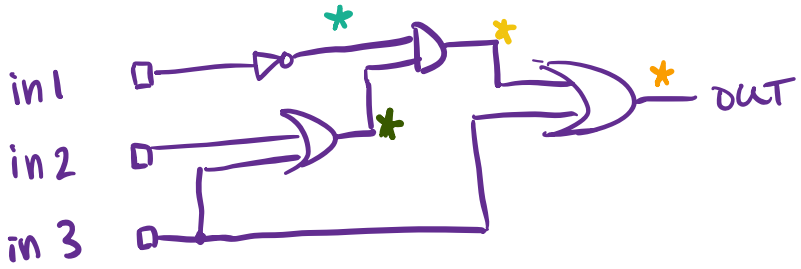
D.




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

ii. Design a circuit that implements the logical expression:

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

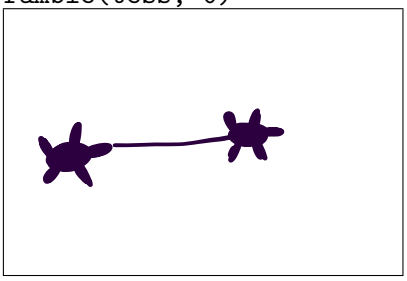


4. (a) Draw the output for the function calls:

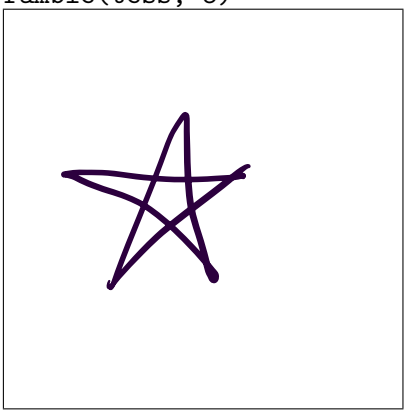
```
import turtle
tess = turtle.Turtle()
tess.shape("turtle")

def ramble(t, side):
    if side == 0:
        t.stamp()
        t.forward(50)
        t.stamp()
    else:
        for i in range(side):
            t.forward(50)
            t.left(720/side)
```

i. `ramble(tess, 0)`



ii. `ramble(tess, 5)`



(b) What is the output:

```
#Another mystery program...
def mystery(num):
     send = chr(num)
     if num < ord("d"):
          send = send + "Y"
     return send

def enigma(letters):
     data = ""
     for x in letters:
          n = ord(x)
          c = "D"
          if n > 100:
            c = mystery(n)
          data = data + c
     return data

word = input("Enter a word: ")
s = enigma(word)
print(s)
```

i. When the user enters: `aa`?

Output:

DD

ii. When the user enters: `cab`?

Output:

DDD

iii. When the user enters: `alice`?

Output:

DiDe

5. Fill in the Python program below. Consider the following pseudocode:

   - Generate a random integer from 1 to 10 (inclusive), call it `x`
   - Print the number in one's complement representation; that is, given a binary string, all 0's become 1's and all 1's become 0's
   - **Example:** 0110 in one's complement representation is 1001

```
#imports the library for generating random numbers
import random

#generates a random integer from 1-10 inclusive
x = random.randrange(1, 11)

#converts the random integer to a binary string
binary = bin(x)[2:]  # sample use:  bin(6)[2:]  == "0110"

#stores the one's complement representation of x
result = ""

#loops through the binary string
for c in binary:
    #if the char is "0", add "1" to result
    if c == "0":
        result += "1"

    #otherwise, add "0" to result
    else:
        result += "0"

#prints x and its one's complement representation
print(x, result)
```

6. Consider the following main function that analyzes star data:

```
import pandas as pd
    def main():
        stars = pd.read_csv("stars.csv")
        avgH = avgRadius(stars, "Hypergiant")
        top3 = topK(stars, "Star color", 3)
```

Define the functions below:

```
def avgRadius(df, starType):
    """
    Takes a DataFrame and a string as input
    Returns the average radius of input value
    First, group by "Star type" then get group starType
    Get the average radius of the group by using the "Radius" column
    """
```

```
stars = df.groupby("Star type").get_group(starType)
return stars["Radius"].mean()
```

```
def topK(df, colName, k):
    """
    Returns the top k values in the given column and DataFrame
    Assume the following:
    - k is a valid integer (will not cause errors)
    - colName is a string that is the name of a column in the DataFrame df
    """
```

```
return df[colName].value_counts()[:K]
```

7. Fill in the Python program below that asks the user for the name of a .png (image) file and **turns the right half of the image red.** The new image should then be displayed to the user.

```
#import the libraries for image processing
import matplotlib.pyplot as plt
import numpy as np

#get user input
infile = input("Enter file name: ")

#read the image file
img = infile

#get the width of the image
width = img.shape[1]

#set the green and blue channels to 0.0
img[:, width//2:, 1:] = 0.0

#set the red channel to 1.0
img[:, width//2:, 0] = 1.0

#load the image into pyplot
plt.imshow(img)

#display the image
plt.show()
```

8. (a) Consider the following MIPS program:

```
ADDI $s0, $zero, 10
ADD $s1, $s0, $s0
ADD $s2, $s1, $s1
ADDI $s3, $s2, 5
```

After the program runs, what is the value stored in:

i. register `$s1`

20

ii. register `$s2`

40

iii. register `$s3`

45

(b) What is the output for a run of this MIPS program:

Output:

LMNO

```
#Loop through four letters:
ADDI $sp, $sp, -5           # Set up stack
ADDI $t0, $zero, 76         # Start $t0 at 76 (L)
ADDI $s2, $zero, 80         # Use to test when you reach 80 (P)
SETUP: SB $t0, 0($sp)       # Next letter in $t0
ADDI $sp, $sp, 1            # Increment the stack
ADDI $t0, $t0, 1            # Increment the letter
BEQ $t0, $s2, DONE          # Jump to done if $t0 == 80
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. Translate the following Python program into a complete C++ program:

```
number = -1
while number < 0:
    number = int(input("Enter a number: "))
if number % 2 == 0:
    print("Even")
else:
    print("Odd")
```

```
//include library for input/output and declare namespace
```

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

```
//main function signature
```

```
int main ()
```

```
{
  //main function body
```

```
int number = -1;
while ( number < 0) {
   cout << "Enter a number";
   cin >> number;
}
if (number % 2 == 0) {
   cout << "Even" << endl;
} else {
   cout << "Odd" << endl;
}
```

```
  return 0;
}
```

10. (a) Write a complete C++ program that prompts the user to enter a time (in 24-hour format) and prints the time of day: `"morning", "afternoon", or "evening".` Morning is any time before 12 P.M. (1200), and evening is any time after 6 P.M. (1800). Make sure to validate the user's input so that the time is between 0 and 2400 (inclusive).

```
//include library for input/output and declare namespace
```

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

```
//main function signature
```

```
int main ()
```

```
{
  //main function body
```

```
int time = -1;
while( time < 0  || time > 2400){
    cout<< "Enter time: ";
    cin >> time;
}
if (time < 1200) {
   cout << "morning\n";
} else if ( time > 1800) {
   cout << "evening\n";
} else {
   cout << "afternoon\n";
}
```

```
  return 0;
}
```

(b) Write a **complete C++ program** that prints the change in population of predator and prey following the Lotka-Volterra model:

$$r = 2 * r - (0.25 * r) * f$$
$$f = 0.95 * f + (0.1 * r) * f$$

Assume that the starting population of prey (rabbits) is 1000 and the starting population of predators (foxes) is 100. Your program should print for the first 10 years: the year, the number of prey, and the number of predators.

```
//include library for input/output and declare namespace
```

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

```
//main function signature
```

```
int main ()
```

```
{
  //calculate and print the predicted population
```

```
double r = 1000.0;
double f = 100.0;
for(int i = 0; i< 10; i++) {
    cout << i << " " << r << " " << f <<endl;
    r= r* 2.0 - (r* 0.25)*f;
    f= f* 0.95 + (r* 0.10)*f;
}
```

```
  return 0;
}
```