Final Exam V2

CSCI 127: Introduction to Computer Science

Hunter College, City University of New York

May 20, 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:
EmpID:
Signature:

If you earn a D in the class and would rather have an F, put an X in this box. ☐
(This will not affect your grade if you earn a C or better.)

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 = "CDEfghE123Ehello"
print(banana.count("e"))
```
Output: 1

ii.
```
B = banana.split("E")
print(B[-1])
```
Output: hello

iii.
```
low = B[0].lower()
print(low)
```
Output: cd

iv.
```
for c in low:
    print(c.upper())
```
Output:
C
D

(b) Consider the contents of the current directory, `Users/Bob`:

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

i. What is the output for:

```
$ ls *c*
```
Output:
banana.csv
carrot.csv
clementine.py

ii. What is the output for:

```
$   mv *.py ./dragonfruit
$   mkdir hello
$   ls
```
Output:
banana.csv dragonfruit
carrot.csv hello

iii. What is the output for:

```
$   cd ./dragonfruit
$   pwd
```
Output: Users/Bob/dragonfruit

2. Complete the Python program below:

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

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

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

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

```
#create an all-black image with a height of 50 and a width of 30
```

```
img = np.zeros((50,30,3))
```

```
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 even rows become the color entered
```

```
else:
    img[0::2, :, 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(0.0, 1.0, 0.0)`
   ☐ black ☐ red ☐ white ☐ gray ☒ green

   ii. Select the SMALLEST binary number:
   ☐ 1011 ☐ 1101 ☐ 0111 ☒ 0010 ☐ 1001

   iii. Select the SMALLEST hexadecimal number:
   ☐ FD ☐ EA ☒ AF ☐ CF ☐ CD

   iv. What is the binary number equivalent to the decimal number 18?
   ☐ 01011 ☒ 10010 ☐ 11100 ☐ 10111 ☐ 10011

   v. What is the hexadecimal number equivalent to the decimal number 60?
   ☐ 34 ☐ 32 ☐ 2C ☒ 3C ☐ 3D

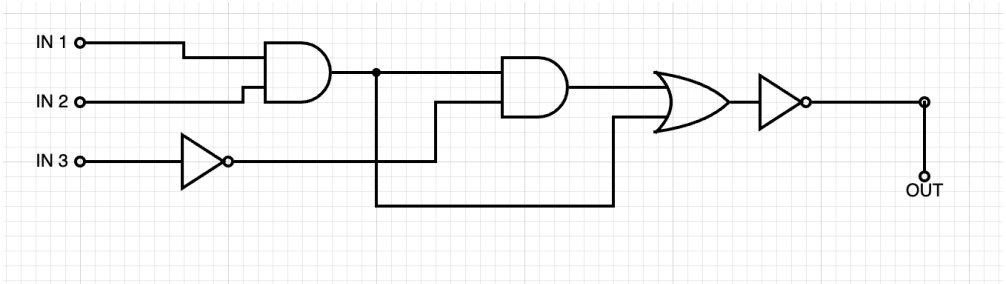
(b) i. What is the value (True/False):

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

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

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

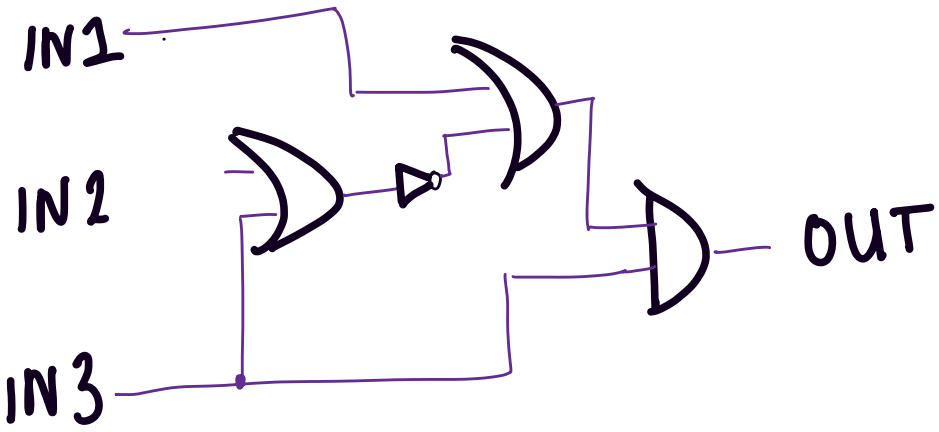
D.




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

ii. Design a circuit that implements the logical expression:

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

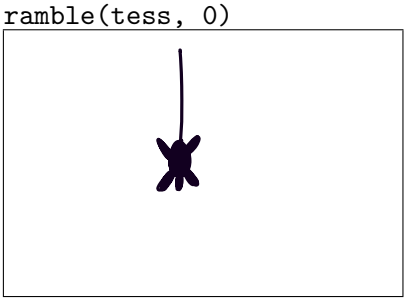


4. (a) Draw the output for the function calls:

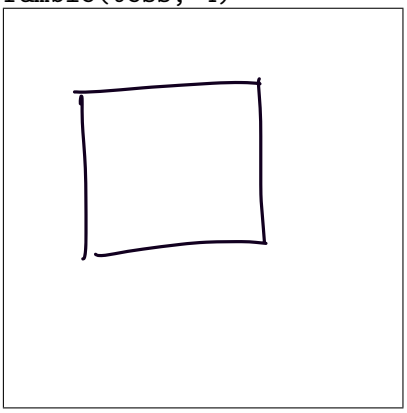
```
import turtle
tess = turtle.Turtle()
tess.shape("turtle")

def ramble(t, side):
    if side == 0:
        t.right(90)
        t.forward(50)
        t.stamp()
    else:
        for i in range(side):
            t.forward(50)
            t.left(360/side)
```

i. `ramble(tess, 0)`



ii. `ramble(tess, 4)`



(b) What is the output:

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

def enigma(letters):
     data = ""
     for x in letters:
          n = ord(x)
          c = "E"
          if n > 101:
            c = mystery(n)
          data += c
     return data

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

i. When the user enters: `cba`?

Output:

EEE

ii. When the user enters: `adult`?

Output:

EEuIt

iii. When the user enters: `child`?

Output:

EhiIE

5. Fill in the Python program below.

```
#imports the library for random numbers and the library for turtles
```

```
import turtle
import random
```

```
colors = ["aliceblue", "burlywood", "cornflowerblue"]
tina = turtle.Turtle()
tina.shape("turtle")

#generates a random integer representing the valid indices of the colors array
rand_color=
```

```
random.randrange(0,3)
```

```
#applies that color to the turtle
tina.color(colors[rand_color])

#generates a random integer representing the number of sides for
#a triangle, square, or pentagon
rand_shape=
```

```
random.randrange(3,6)
```

```
#for-loop to draw the shape
```

```
for i in range(rand_shape):
    #move tina forward 50 steps
```

```
    tina.forward(50)
    #turn tina left the correct number of degrees
```

```
    tina.left(360/rand_shape)
```

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

```
import pandas as pd
    def main():
        trees = pd.read_csv("trees.csv")
        avgOak = avgHeight(trees, "Oak")
        topTrees = topK(trees, "circumference")
```

Define the functions below:

```
def avgHeight(data, species):
    """
    Takes a DataFrame and a string as input
    First, group by "Species" then get group species
    Return the average height of the group by using the "Height" column
    """
```

```
groups = data.groupby('Species')
groups = groups.get_groups(species)
return groups['Height'].mean()
```

```
def topK(data, colName):
    """
    Takes a DataFrame and a string as input
    Asks the user for an integer value, k
    Returns the top k values in the given column and DataFrame
    """
```

```
K= int(input("Enter value:"))
return data[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 top half of the image blue.** The new image should then be displayed to the user.

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

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

#read the image file
img = plt.imread(infile)

#get the height of the image
height = img.shape[0]

#set the red and green channels to 0.0
img[:height//2, :, 0:2] = 0.0

#set the blue channel to 1.0
img[:height//2, :, 2] = 1.0

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

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

8. (a) Consider the following MIPS program:

```
ADDI $s1, $zero, 6
ADD $s2, $s1, $s1
ADDI $s2, $s2, 1
ADDI $s3, $s2, 5
```

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

i. register `$s1`

6

ii. register `$s2`

13

iii. register `$s3`

18

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

Output:

PQRST

```
#Loop through five letters:
ADDI $sp, $sp, -6           # Set up stack
ADDI $t0, $zero, 80         # Start $t0 at 80 (P)
ADDI $s2, $zero, 85         # Use to test when you reach 85 (U)
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 == 85
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, -6           # 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:

```
dividend = float(input("Enter a number: "))
divisor = float(input("Enter a number: "))

while divisor != 0:
    print("The quotient is: ", dividend/divisor)
    divisor = float(input("Enter a number: "))

print("Cannot divide by zero")
```

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

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

```
//main function signature
```

```
int main ()
```

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

```
float dividend, divisor;
cout << "Enter a number: ";
cin >> dividend;
cout << "Enter a number: ";
cin >> divisor;
while (divisor != 0) {
    cout << " The quotient is: " << dividend / divisor << "\n";
    cout << "Enter a number: ";
    cin >> divisor;
}
cout << " Cannot divide by zero \n";
```

```
  return 0;
}
```

10. (a) Write a complete C++ program that prompts the user to enter a number representing an exam grade. The program should print `"Pass"` when the value is greater than or equal to `60` and `"Fail"` otherwise. Validate the user's input so that the number entered is between 0 and 100 inclusive.

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

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

```
//main function signature
```

```
int main ()
```

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

```
int grade = -1;
while ( grade < 0 || grade > 100){
   cout << " Enter score ";
   cin >> grade;
}
if( grade >= 60){
   cout << " Pass \n";
} else {
   cout << " Fail \n";
}
```

```
  return 0;
}
```

(b) Write a complete C++ program that prints the first 10 numbers of the Fibonacci sequence using a for-loop. Use the following pseudocode to implement your main function:

```
1. Declare three integers: a, b, and c. Initialize a to 0 and b to 1.
2. Print out a and then b, separated by newline characters
3. For i = 2, 3, 4, ... , 9:
     c = a + b
     Print c followed by a newline
     a = b
     b = c
```

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

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

```
//main function signature
```

```
int main ()
```

```
{
  //calculate and print first 10 Fibonacci numbers
```

```
int a = 0;
int b = 1;
int c;

cout << a << "\n" << b << "\n";

for (int i = 2; i < 10; i++) {
    c = a + b;
    cout << c << "\n";
    a = b;
    b = c;
}
```

```
  return 0;
}
```