Final Exam

CSCI 127: Introduction to Computer Science

Hunter College, City University of New York

Version 02

December 16, 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 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, smartwatch, 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:

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]*

1. (a) **What will the following Python code print?** *Note that each section is run sequentially, so the commands from part i will affect part ii and so on.*

i.

```
apples = "Granny Smith%Gala"
print(apples.find("%"))
```

Output:

12

ii.

```
apples += "%Honeycrisp%"
aList = apples.split("%")
print(len(aList))
```

Output:

4

iii.

```
up = aList[2].upper()
print(ord(up[0]))
```

Output:

72

iv.

```
for apple in aList:
    print(apple.count("n"))
```

Output:

2
0
1

(b) Consider the contents of the current directory, **/Desktop/fruit**:

```
lemon.csv  lime.txt   orange.py  clementine.py   citrus
```

Note that each section is run sequentially, so the commands from part i will affect part ii and so on.

i. What is the output for:

```
$ ls *o*
```

Output:

lemon.csv orange.py

ii. What is the output for:

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

Output:

lemon.csv lime.txt
citrus

iii. What is the output for:

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

Output:

Desktop

2. Complete the Python program below that creates an image of a topographic map based on elevation levels.

   First, ask the user to enter a value representing an amount of blue. This value will be in the range [0.0, 1.0].

   Then, color the pixels of the image as follows:

   If the elevation is less than or equal to 0, color the pixel the amount of blue the user specified.

   If the elevation is greater than 0 but less than 30, color the pixel black.

   Otherwise, the pixel should be colored white.

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

elevations = np.loadtxt("elevationsNYC.txt")

mapShape = elevations.shape + (3,)
topoMap = np.zeros(mapShape)

#YOUR CODE HERE
blue = float(input("Enter blue:"))

for i in range(mapShape[0]):
    for j in range(mapShape[1]):

        if elevations[i,j]  <=  0:
            topoMap[i,j,2] = blue
        elif elevations[i,j] < 30:
            topoMap[i,j,:] = 0.0
        else:
            topoMap[i,j,:] = 1.0
```

3. (a) Select the correct option.

i. What color is tina after this command? `tina.color("#555555")`
□ green ■ gray □ white □ red □ blue

ii. Select the SMALLEST binary number:
■ 0011 □ 0100 □ 0111 □ 1010 □ 1001

iii. Select the LARGEST hexadecimal number:
■ B6 □ 7A □ 9F □ 91 □ B3

iv. What is the binary number equivalent to the decimal number 28?
□ 01111 □ 11011 ■ 11100 □ 10111 □ 10111

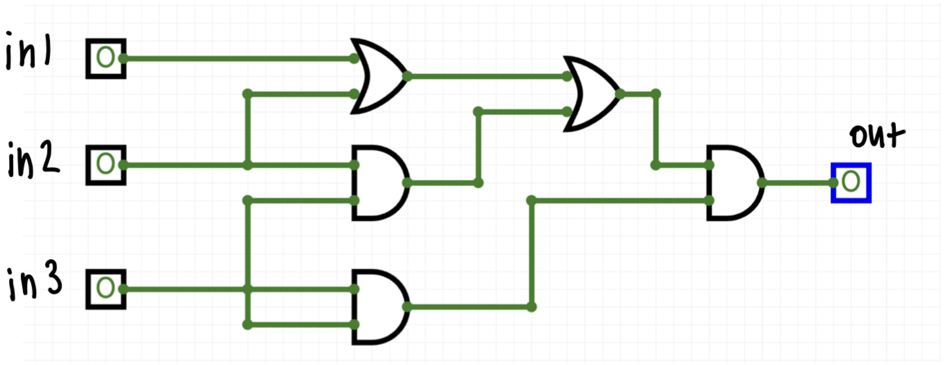
v. What is the hexadecimal number equivalent to the decimal number 28?
□ 1A □ 1B ■ 1C □ 1D □ 1E

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

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

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

C.
```
in1 = False
in2 = True or in1
in3 = (in1 or in2) and True
out = in2 and in3
```
out = True

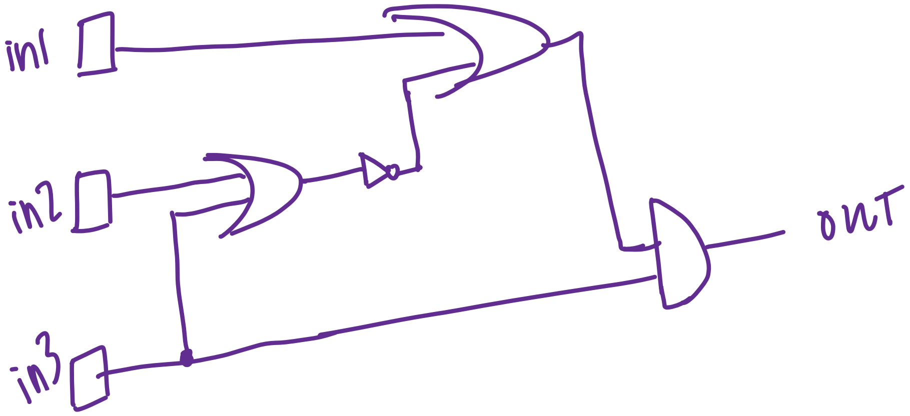


(c) Write the logical expression that represents the circuit shown above.

out = (in3 and in3) and ((in2 and in3) or (in1 or in2))

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

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



4. Write a Python program to make a turtle walk 300 times. Each “walk” is 50 steps forward. The turtle should turn right `[ 90, 180, 270, 360 ]` degrees (chosen randomly) at the beginning of each walk.

```
import turtle
import random

tina= turtle.Turtle()


for i in range(300):
    deg= random.randrange(90,361,90)
    tina.right(deg)
    tina.Forward(50)
```

5. Consider the following dataset:

TEMP	LUM	RADIUS	COLOR	TYPE
400	0.0005	79	Red	Brown Dwarf
1482	0.0024	55	Orange	Brown Dwarf
18489	59302134	65723	Blue	Hypergiant
14553	47821947	102.34	Blue	Hypergiant

Assume this data is stored in `stars.csv`

(a) Write a Python program that finds the **coldest** star in the dataset and prints its temperature in Fahrenheit. The temperature data is originally in Kelvin. The formula to convert Kelvin to Fahrenheit is $F = \frac{9}{5}(K - 273.15) + 32$, where K is the degrees in Kelvin and F is the degrees in Fahrenheit.

```
import pandas as pd
df= pd.read_csv("stars.csv")
coldest= df["TEMP"].min()
F= 9/5 * (coldest - 273.15)+32
print("Coldest star", F)
```

(b) Write a Python program that prints the average temperature of stars that are `"Red"` in color.

```
import pandas as pd
df= pd.read_csv("stars.csv")
reds = df.groupby("COLOR"). get_group("Red")
avgTemp= reds["TEMP"].mean()
print("Avg temp of red stars:", avgTemp )
```

6. Consider the following main function:

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

def main():
    # saves an all-green image with a height of 50 and width of 100
    customImg(50,100,1)
```

Define the function below:

```
def customImg(numRows, numCols, color):
#   1. Create an image with a height of numRows and a width of numCols
#   2. If the color parameter is not 0, 1, or 2, print "Invalid color channel."
#   3. Otherwise, modify the image such that all pixels become the color entered
#   4. Save the image as a file called "myPic.png"
```

```
#1. Create image
img = np.zeros((numRows, numCols, 3))
#2. Check color parameter
if color != 0 or color != 1 or color != 2 :
    print("Invalid color channel")
#3. Modify image
else:
    img[:,:, color] = 1.0
#4. Save image
plt.imsave("myPic.png", img )
```

7. Write a complete Python program that asks the user for the name of an image file and prints the number of mostly blue pixels in that image. A pixel is mostly blue if the amount of red and green are both below 0.25 and the amount of blue is above 0.75.

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

filename = input("Enter filename:")
img= plt.imread(filename)

count = 0

for i in range(img.shape[0]):
    for j in range(img.shape[1]):
        r, g, b = (img[i,j,0], img[i,j,1], img[i,j,2])

        if r< 0.25 and g < 0.25 and b > 0.75:
            count += 1

print("Number of mastly blue pixels:", count)
```

8. (a) Consider the following MIPS program:

```
ADDI $s1, $zero, 20
ADD $s2, $s1, $s1
ADDI $s2, $s2, 100
ADDI $s3, $s2, 200
```

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

i. register $s1

20

ii. register $s2

140

iii. register $s3

340

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

Output:

CDEF

```
#Loop through four letters:
ADDI $sp, $sp, -5           # Set up stack
ADDI $t0, $zero, 67         # Start $t0 at 67 (C)
ADDI $s2, $zero, 71         # Use to test when you reach 71 (G)
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 == 71
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:

```
myString = input("Enter a string: ")
numChars = len(myString)
idx = numChars // 2
print("Middle:")
if numChars % 2 == 1:
    print(idx)
else:
    print(idx-1, idx)
```

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

int main() {
```

```
cout << "Enter string";
string myString;
cin >> myString;
int numChars;
numChars = myString.length();
int idx = num chars / 2;
cout << "Middle \n";
if (numChars % 2 == 1) {
   cout << idx << "\n";
}else {
  cout << idx-1 << "  " << idx << "\n";
}
```

```
    return 0;
}
```

10. Write a program in C++ that outputs the numbers from 1 to a user-specified integer. The program should follow these rules:

```
For each number in the range,
a) If the number is divisible by both 4 and 8, print "FooBar"
b) If the number is divisible by 4, print "Foo"
c) If the number is divisible by 8, print "Bar"
d) For all other numbers, print the number itself
```

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

int main() {
  int N;
  cin >> N;

  for (int i = 1; i <= N; i++) {
```

```
if(i%4==0 && i%8==0){
   cout << "FooBar ";
} else if ( i%4 == 0 ) {
   cout << "Foo ";
} else if ( i %8 == 0 ) {
   cout << "Bar ";
} else {
   cout << i;
}
```

```
  } //loop end

  return 0;
}
```