Row:	Seat:

Mock Final Exam

CSci 127: Introduction to Computer Science

Hunter College, City University of New York

May 16, 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.

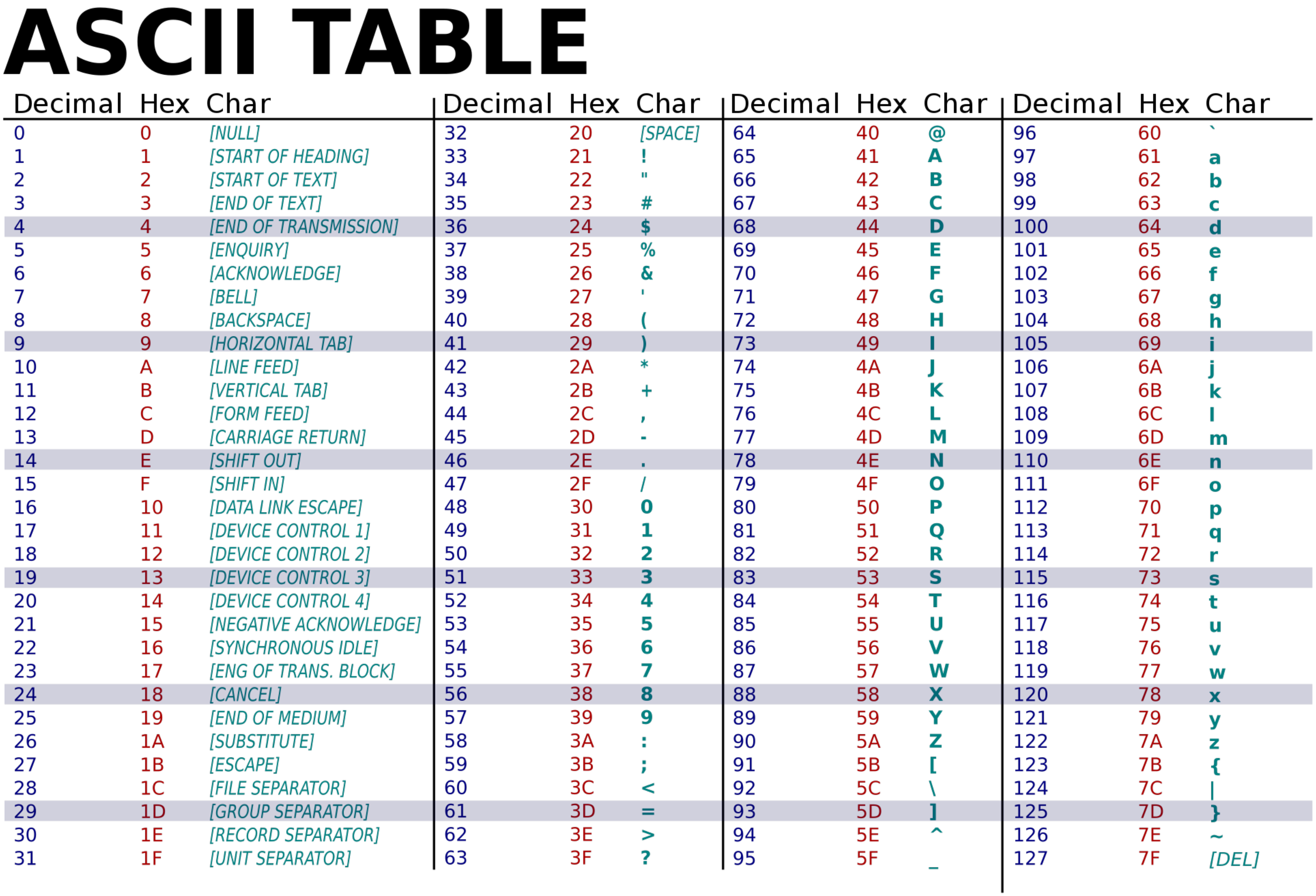
I understand that all cases of academic dishonesty will be reported to the Dean of Students and will result in sanctions.

Name:

EMPLID: □□□□□□□□

Email:

Signature:



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) Fill in the code below to produce the Output on the right:

```
seasons = "Spring,Summer,Autumn,Winter"
```

i.
```
autumn_winter = 
for s in autumn_winter:
    print(                    )
```

Output:

```
autumn
winter
```

ii.
```
spring_autumn = 
for s in spring_autumn:
    print(                    )
```

Output:

```
SPRING
AUTUMN
```

(b) Consider the following shell commands:

```
$ pwd
/usr/student
$ ls
hello.csv grades.csv  test.py  hello.py
```

i. What is the output for:

```
$ mkdir data
$ mv *csv data
$ cd data
$ ls
```

Output:

ii. What is the output for:

```
$ cd ../
$ ls -l | grep hello | wc -l
```

Output:

iii. What is the output for:

```
$ ls | grep test
```

Output:

2. (a) Select the color corresponding to the rgb values below:

   i. `rgb = (65, 65, 65)`
   □ black □ red □ white □ gray □ blue

   ii. `rgb = "#0000AB"`
   □ black □ red □ white □ gray □ blue

   iii. `rgb = (255, 255, 255)`
   □ black □ red □ white □ gray □ blue

   iv. What is the binary number equivalent of decimal number 54?

   `Decimal 54 = Binary` □□□□□□

   v. What is the Decimal number equivalent to Hexadecimal 2F?

   `Hexadecmal 2F = Decimal` □□□

   (b) Given the list `fruits` below, fill in the code to produce the Output on the right:

```
fruits = ["orange", "banana", "apple", "cherry", "strawberry"]
```

i.

```
for i in range( □ ):
    for j in range( □ ):
        print(fruits[j]□, end=" ")
```

Output:

```
o b a c s
o b a c s
```

ii.

```
for j in range( □ , □ , □ ):
    print(fruits[j]□, end=" ")
```

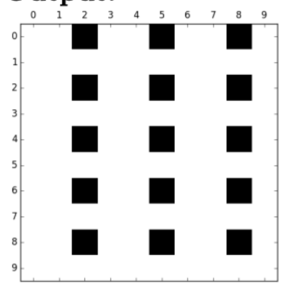
Output:

```
e e y
```

iii.

```
import numpy as np
import matplotlib.pyplot as plt
img = np.ones( (10,10,3) )
img[□ , □, □ ] = 0
plt.imshow(img)
plt.show()
```

Output:



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

i.

```
in1 = False
in2 = True
out = not in1 and in2
```

□ True □ False

ii.

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

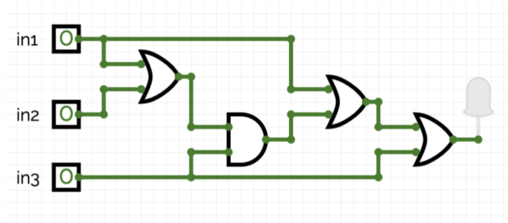
□ True □ False

iii.

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

□ True □ False

iv.



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

□ True □ False

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

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

4. Consider the following functions:

```
def hello(chris, amy):
    amanda = 0
    for num in chris:
        if frog(num, amy):
           amanda += 2

    return amanda
```

```
def frog(a, b):
    return a > b

def main():
    mylist = [1, 6, 5, -3, 7]
    print(hello(mylist, 2))
```

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

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

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

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

i. **Output:**

5. Design an algorithm that asks the user for the name of a text file containing a grid of numbers and loads it into a 2D array of integers (think like an image without the color channel), then outputs the index (row, col) of the LARGEST number in the array.

Libraries:

Input:

Output:

Design Pattern:

□ Search □ Find Min □ Find Max □ Find All

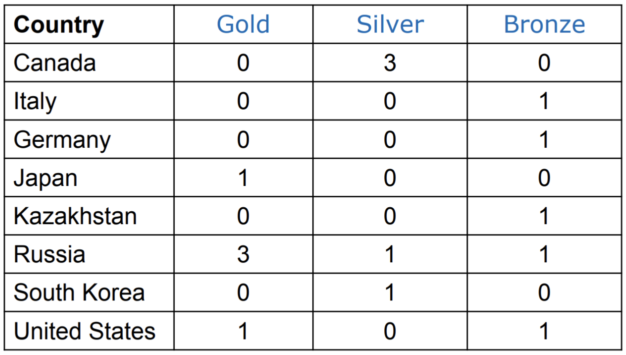
Principal Mechanisms (select all that apply):

□ Single Loop □ Nested Loop □ Conditional (if/else) statement
□ Indexing / Slicing □ `split()` □ `groupby()`

Process (as a concise and precise LIST OF STEPS / pseudocode):

(Assume libraries have already been imported.)

6. Consider the `medalcount.csv` dataset that reports the medal count for skating at the 2014 Winter Olympics. A snapshot is given in the image below:



Country	Gold	Silver	Bronze
Canada	0	3	0
Italy	0	0	1
Germany	0	0	1
Japan	1	0	0
Kazakhstan	0	0	1
Russia	3	1	1
South Korea	0	1	0
United States	1	0	1

Fill in the Python program below:

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

```
#Read input data into data frame:

df =
```

```
#Create a new column that has a total medal count for each country
```

7. Write a **complete Python program** that prompts the user for the name of an .png (image) file and prints the fraction of pixels that are grayscale, or a shade of gray. Recall that a pixel is a shade of gray if the red, green, and blue values are all equal.

8. (a) What does the MIPS program below print:

   Output:

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

```
□ ADDI $sp, $sp, -6

□ ADDI $s3, $zero, 1

□ ADDI $t0, $zero, 65

□ ADDI $s2, $zero, 5

□ 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 # Set up stack to print

□ 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(                           ){
       cout << i*2 << endl;
    }
    return 0;
}
```

Output:

```
200
400
600
800
```

(b)

```
#include <iostream>
using namespace std;
int main()
{
    int count = 20;
    int num = 10;

    while(                          ){
        cout << count << " " << num << endl;
        count -= 50;
        num -= 5;
    }
    return 0;
}
```

Output:

```
200 100
150 95
100 90
```

(c)

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

    for(                           ){
         cout << "Hello" << endl;
    }
    return 0;
}
```

Output:

```
Hello
Hello
Hello
Hello
Hello
```

10. (a) Translate the following python program into a **complete C++ program**:

```
for i in range(97,113,3):
  for j in range(i,60,-4):
    print(i," ",j)
```