Row:	SEAT:

FINAL EXAM, VERSION 2 CSci 127: Introduction to Computer Science Hunter College, City University of New York

Spring 2025

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 $1/2" \ge 11"$ 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 Stud	ents	and v	will r	esult	in sa	anctio	ons.		
Name:									
EmpID:									
Email:									
Signature:									

ASCII TABLE Decimal Hex Char Decimal He

Decimal	нех	Char	Decin
0	0	[NULL]	32
1	1	[START OF HEADING]	33
2	2	[START OF TEXT]	34
3	3	[END OF TEXT]	35
4	4	[END OF TRANSMISSION]	36
5	5	[ENQUIRY]	37
6	6	[ACKNOWLEDGE]	38
7	7	[BELL]	39
8	8	[BACKSPACE]	40
9	9	[HORIZONTAL TAB]	41
10	А	[LINE FEED]	42
11	В	[VERTICAL TAB]	43
12	С	[FORM FEED]	44
13	D	[CARRIAGE RETURN]	45
14	E	[SHIFT OUT]	46
15	F	[SHIFT IN]	47
16	10	[DATA LINK ESCAPE]	48
17	11	[DEVICE CONTROL 1]	49
18	12	[DEVICE CONTROL 2]	50
19	13	[DEVICE CONTROL 3]	51
20	14	[DEVICE CONTROL 4]	52
21	15	[NEGATIVE ACKNOWLEDGE]	53
22	16	[SYNCHRONOUS IDLE]	54
23	17	[ENG OF TRANS. BLOCK]	55
24	18	[CANCEL]	56
25	19	[END OF MEDIUM]	57
26	1A	[SUBSTITUTE]	58
27	1B	[ESCAPE]	59
28	1C	[FILE SEPARATOR]	60
29	1D	[GROUP SEPARATOR]	61
30	1E	[RECORD SEPARATOR]	62
31	1F	[UNIT SEPARATOR]	63

 $(From\ wikipedia\ commons)$



(inom tratintablogomeoin)					
Furtles: Let t be	a turtle.				
Function	Description				
t.forward(x)	Move turtle forward x steps.				
t.backward(x)	Move turtle backward \mathbf{x} steps.				
t.left(x)/t.right(x)	Turn turtle left/right \mathbf{x} degrees.				
t.penup()/t.pendown()	Lift turtle's pen up/down.				
t.stamp()	Stamp at current location.				
t.goto(x,y)	Move turtle to (x,y).				
<pre>t.left(x)/t.right(x) t.penup()/t.pendown() t.stamp() t.goto(x,y)</pre>	Turn turtle left/right x degrees. Lift turtle's pen up/down. Stamp at current location. Move turtle to (x,y).				

String Methods:	Let s be a string.
Function	Description
len(s)	Returns the length of s .
s.lower()	Returns \mathbf{s} as lower case characters.
s.upper()	Returns \mathbf{s} as upper case characters.
s.find(t)	Returns index of t in s (-1 not found).
s.split(d)	Splits s into list of strings on d.
s.join[lst]	Joins 1st into a string, by s.

nal	Hex	Char	Decimal	Hex	Char	Decimal	Hex	Char
	20	[SPACE]	64	40	0	96	60	×
	21	1	65	41	Α	97	61	а
	22		66	42	В	98	62	b
	23	#	67	43	С	99	63	с
	24	\$	68	44	D	100	64	d
	25	%	69	45	E	101	65	е
	26	&	70	46	F	102	66	f
	27	1.00	71	47	G	103	67	g
	28	(72	48	H	104	68	ĥ
	29)	73	49	1	105	69	i.
	2A	*	74	4A	J	106	6A	j
	2B	+	75	4B	κ	107	6B	k
	2C	,	76	4C	L	108	6C	1
	2D		77	4D	м	109	6D	m
	2E	100 C	78	4E	Ν	110	6E	n
	2F	1	79	4F	0	111	6F	0
	30	0	80	50	Ρ	112	70	р
	31	1	81	51	Q	113	71	q
	32	2	82	52	R	114	72	r
	33	3	83	53	S	115	73	S
	34	4	84	54	т	116	74	t
	35	5	85	55	U	117	75	u
	36	6	86	56	V	118	76	v
	37	7	87	57	W	119	77	w
	38	8	88	58	X	120	78	x
	39	9	89	59	Y	121	79	У
	3A	÷	90	5A	Z	122	7A	z
	3B	;	91	5B	[123	7B	{
	3C	<	92	5C	1	124	7C	
	3D	=	93	5D]	125	7D	}
	3E	>	94	5E	^	126	7E	~
	3F	?	95	5F	_	127	7F	[DEL]

Pandas: Let df be a DataFrame, s a Series, & pd the Pandas package.

i allaas. Det ui be a Data	rame, s a beries, & pu the randas package.
Function	Description
pd.read_csv(fn)	Returns a DataFrame with file fn.
pd.to_csv(fn)	Writes df to fn.
pd.DataFrame(d)	Returns DataFrame built from dictionary d.
df[col]	Returns col column as a Series.
df[[col1,col2]]	Returns DataFrame with col1 & col2.
df.columns	List of column names of df.
df.head(n)/df.tail(n)	First/last n lines of df.
df.plot(x=col)	Returns a figure with col as x-axis
fig.savefig(fn)	Writes fig to fn.
<pre>s.min()/s.max()/s.mean()</pre>	Returns \min/\max /average of s .
<pre>s.value_counts()</pre>	Counts $\#$ times each value occurs.
df.groupby(col)	Groups df by values in col.
Plotly Express: Let p	t be the Plotly Express package.
Function	Description
longitude	Degrees east/west from -180 to 180.
latitude	Degrees north/south from -90 to 90 .
$px.scatter_geo(df,)$	Returns outline map as fig. Keywords args:
	<pre>lon,lat,size,hover_name,projection,title.</pre>
$px.scatter_map(df,)$	Returns tiled map as fig. Keywords args:
	<pre>lon,lat,size,hover_name,title,zoom.</pre>
fig.show()	Displays map on browser.
_fig.write_html(fn)	Writes fig to fn.
MIPS: Let rs, rt, & rd b	e registers.
Function	Description
ADD rd, rs, rt	Adds values of rs and rt and stores in rd.
ADDI rd, rs, imm	Adds values of rs and imm and stores in rd.
SUB rd, rs, rt	Subtracts values of rs and rt and stores in rd.
BEQ rs, rt, target	If registers rs == rt, jump to target.
JUMP target	Jump to target.
UNIX:	
Function	Description
ls / ls -l / ls *.py	Lists files /lists long/lists matching pattern.
cp x y / mv x y	Copies/renames file x to file y.
pwd	Prints path to current directory.
mkdir x	Creates directory called x.
cd/ / cd /usr/bin	Changes directory via relative/absolute path.
ls wc -c / ls grep pat	Uses pipes to count $\#$ of files/match pat

SCRATCH PAPER

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

```
num_s = "two twenty-one thirty-two three twenty"
nums = num_s.split(" ")
print(nums[-1])
count = num_s.count("-")
print("List has", count, "two-parts.")
two_nums = [n for n in nums if "-" in n]
print(two_nums)
ones = ["zero", "one", "two", "three", "four"]
tens = ["", "", "twenty", "thirty", "forty"]
for num in two_nums:
    places = num.split("-")
    dec = ones.index(places[1])+\
        tens.index(places[0])*10
print(num, "=", dec)
```

(b) Consider the following shell commands:

```
$ ls
code_p1.png code_p2.png exams key.pdf
$ file exams
exams: directory
$ pwd
/tmp/final/ver2
```

Assuming the commands below are run sequentially, what is the output after each has run:

i. \$ mv key.pdf exams
\$ ls
\$ cd exams
\$ cp key.pdf key_2.pdf
\$ ls
\$ mkdir answers
\$ cd answers
\$ pwd

Output:

Output:

Output:

Output:

iv. \$ cd ../../ \$ ls | grep c 2. (a) Fill in the missing values in the table:

Decimal	Binary	Hexadecimal
5		5
	1100	С
33	100001	
253	11111101	

(b) Fill in the missing code to make the image: import turtle turtle.colormode(255)

```
#Create turtle named tad:
```

tad =

tad.sł	nape	("turtle")		
#Move	tad	backwards	100	steps:

```
#For 0,10,20,...,250
```

for i

tad.1	forward	1(10))				
tad.p	pensize	e(i))				
#Set	color	to	red=i,	no	blue,	no	green:

(c) Consider the code:

```
import pandas as pd
csvFile = input('Enter CSV file name: ')
recipe = pds.read_csv(csvFile)
recipe["Amount'] = 2*recipe["Amount"]
print(recipe)
```

- ii. Box the code above and mark line with (ii) that caused this error: line 4: recipe["Amount'] = 2*recipe["Amount"]

```
SyntaxError: unterminated string literal (detected at line 4)
Write the code that would fix the error:
```

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

in1 = False
i. in2 = True
out = in1 or in2
in1 = False
ii. in2 = False
out = in1 or (not in1 and not in2)





in1 = False

(b) Fill in the values to yield the output:



out = False

(c) Design a circuit that implements the logical expression:

out = (in1 and in2) or not in2

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



- 4. (a) Draw the output for the function calls:
- i. ramble(tiago,0)

```
import turtle
tiago = turtle.Turtle()
tiago.shape("circle")

def ramble(t,side):
    if side < 3:
        t.stamp()
    else:
        for i in range(side):
            t.forward(side*10)
            t.left(360/side)
        ramble(t,side-1)</pre>
```



def v2(jaime, lily):
 if jaime + lily > 10:

def start():
 gwenael = 8
 karen = 10
 ryan = v2(gwenael,karen)
 return ryan

(b) For the following code:

i. What are the formal parameters for v2():

ii. What are the formal parameters for start():

iii. What value does start() return:

if jaime + lily > 10: return lily else: return -1

1		

5. Write a function most_common() that takes a string, converts it to lower case, and returns the character that occurs most in the string. If there is a tie for most occurrences, return the first alphabetically. For example:

most_common("Mihi cura futuri")

would return i since both i and u occur the most times (3), but i is first alphabetically.

Libraries:	
Input:	
Output:	

Design Pattern:

 \Box Accumulator \Box Max/Min \Box Finding Duplicates \Box Searching

Principal Mechanisms (select all that apply):

\Box Single Loop	\Box Nested Loop	\Box Conditional (if/else)	\Box Recursion
\Box Indexing/slicing	\Box Dictionary	\Box List Comprehension	\Box Regular Expressions
Process (as a concis	e and precise LIS	Γ OF STEPS / pseudoco	de):

(Assume libraries have already been imported.)

6. Fill in for the code below to create an interactive map, based on housing data. Your program should ask the user for the input and output file names. It should read in the CSV file and create a new column that sums up the number of studio, 1-bedroom, and 2+ bedroom apartments in a single new column, Total Units. A interactive HTML map, based on the DataFrame entries, is saved to the specified outfile.

```
#Import pandas and plotly express libraries as pd and px:
#Ask user for file name:
file_name =
#Read in the file to a DataFrame:
df =
#Make a new column that sums up "Studio", "1Bed", "2+Bed" columns:
df["Total Units"] =
#Use df to make a scatter_map: columns: "latitude" and "longitude" for location,
# "Project Name" for hover_name, & "Total Units" for size:
fig =
#Ask user for output file name:
html_file =
#Save the map as an html file to name given by user:
```

7. Write a **complete Python program** that

- asks the user for the name of a png file and
- prints the number of pixels that are bright red (the fraction of red is above 0.75 and the fraction of green, and the fraction of blue are below 0.25).

8. (a) Consider the following MIPS program:

```
ADDI $s0, $zero, 1
ADD $s1, $s0, $s0
ADD $s2, $s1, $s0
ADD $s3, $s2, $s0
```

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

\$s1 register	\$s2 register	\$s3 register

(b) Consider the MIPS code:

```
ADDI $sp, $sp, -4
1
   ADDI $t0, $zero, 83
\mathbf{2}
   ADDI $s2, $zero, 86
3
   SETUP: SB $t0, 0($sp)
4
   ADDI $sp, $sp, 1
\mathbf{5}
   ADDI $t0, $t0, 1
6
   BEQ $t0, $s2, DONE
\overline{7}
   J SETUP
8
   DONE: ADDI $t0, $zero, 0
9
   SB $t0, 0($sp)
10
   ADDI $sp, $sp, -3
^{11}
   ADDI $v0, $zero, 4
12
   ADDI $a0, $sp, 0
13
   syscall
14
```

i) How many characters are printed?	
ii) What is the first character printed?	
iii) What is the whole message printed?	
iv) Detail the changes needed to the code to print the message in reverse:	

9. (a) What is the output

```
//Neil deGrasse Tyson
#include <iostream>
using namespace std;
int main()
{
    cout << "There is no "
        << "greater educ";
    cout << "ation\nthan one ";
    cout << "that is self-driven."
        << endl;
}</pre>
```

Output:

(b) What is the output:

```
#include <iostream>
using namespace std;
int main()
{
    cout << "Begin" << endl;
    int x = 2;
    while (x > 0) {
        cout << "Again\n";
        x--;
    }
    cout << "End" << endl;
    return 0;
}</pre>
```



(c) What is the output:

```
#include <iostream>
using namespace std;
int main(){
   for (int i=0; i<4; i++){
      for(int j=0; j<4; j++){
        if ( j % 2 == 0)
           cout<<"+";
        else
           cout<<"-";
      }
      cout << endl;
   }
   return 0;
}</pre>
```

Output:

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

```
C++
Python program:
num = 1
while (num > 100) or (num % 2 == 1):
    num = int(input("Enter small even #: "))
print("Your number:", num)
```

```
C++ program:
```

(b) Write a C++ program that will ask for the time in 24 hour format (e.g. 2034 is 8:34pm) and, prints out "Good Morning" if it is before noon (e.g. 1200), "Good Evening" if it after 6pm (e.g. 1800), and otherwise print "Good Afternoon." A sample run:

Enter time: 1345 Good Afternoon