Row:	Seat:

Final Exam, Version 3
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" 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.**

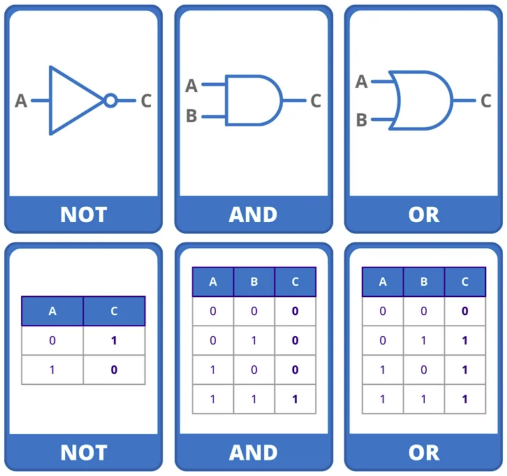
Hunter College regards acts of academic dishonesty (e.g., plagiarism, cheating on examinations, obtaining unfair advantage, and falsification of records and official documents) as serious offenses against the values of intellectual honesty. The College is committed to enforcing the CUNY Policy on Academic 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:
EmpID:
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]*

(From wikipedia commons)



A	C
0	1
1	0

A	B	C
0	0	0
0	1	0
1	0	0
1	1	1

A	B	C
0	0	0
0	1	1
1	0	1
1	1	1

(from truthtablegen.com)

Turtles: Let `t` be a turtle.

Function	Description
`t.forward(x)`	Move turtle forward `x` steps.
`t.backward(x)`	Move turtle backward `x` steps.
`t.left(x)`/`t.right(x)`	Turn turtle left/right `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)`.

String Methods: Let `s` be a string.

Function	Description
`len(s)`	Returns the length of `s`.
`s.lower()`	Returns `s` as lower case characters.
`s.upper()`	Returns `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 `lst` into a string, by `s`.

Pandas: Let `df` be a DataFrame, `s` a Series, & `pd` the Pandas 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`.
`s.min()`/`s.max()`/`s.mean()`	Returns min/max/average of `s`.
`s.value_counts()`	Counts # times each value occurs.
`df.groupby(col)`	Groups `df` by values in `col`.

Plotly Express: Let `px` 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: `lon,lat,size,hover_name,projection,title`.
px.scatter_map(df,...)	Returns tiled map as fig. Keywords args: `lon,lat,size,hover_name,title,zoom`.
fig.show()	Displays map on browser.
fig.write_html(fn)	Writes `fig` to `fn`.

MIPS: Let `rs`, `rt`, & `rd` be 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.
`echo "message"`	Displays message
`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 = "one,twenty,thirty three,four,twenty two"
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)
```

Output:

(b) Consider the following shell commands:

```
$ ls
p1_out.png        p10_out.png        images                lect1.pdf
$ file images
images: directory
$ pwd
/tmp/final/ver3
```

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

i.
```
$ mv lect1.pdf l1.pdf
$ ls
```
Output:

ii.
```
$ mv *.png images
$ ls
```
Output:

iii.
```
$ cd images
$ pwd
```
Output:

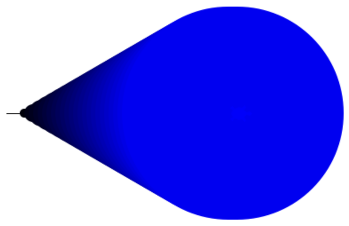
iv.
```
$ echo "Num is:"
$ ls p* | wc -l
```
Output:

2. (a) Fill in the missing values in the table:

Decimal	Binary	Hexadecimal
5		5
	1100	C
33	100001	
253	11111101	

(b) Fill in the missing code to make the image:

```
import turtle
turtle.colormode(255)
#Create turtle named tia:

tia = 

tia.shape("turtle")
#Move tia backwards 100 steps:


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

for i                          :

    tia.forward(10)
    tia.pensize(i)
    #Set color to blue=i, no red, no green:

```



(c) Consider the code:

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

i. **Circle** the code above and mark line with **(i)** that caused this error:

```
csvFile = input("Enter CSV file name: ')
                ^
SyntaxError: unterminated string literal (detected at line 2)
```

Write the code that would fix the error:

ii. **Box** the code above and mark line with **(ii)** that caused this error:

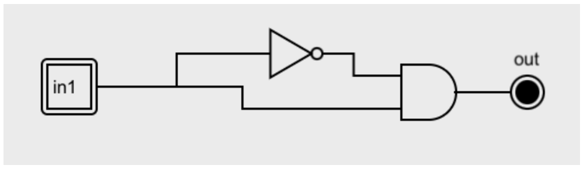
```
line 3: recipe = pds.read_csv(csvFile)
                 ^^^
NameError: name 'pds' is not defined. Did you mean: 'pd'?
```

Write the code that would fix the error:

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

   i.
```
in1 = True
in2 = True
out = in1 or in2
```
   out =

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

   iii. 

   out =

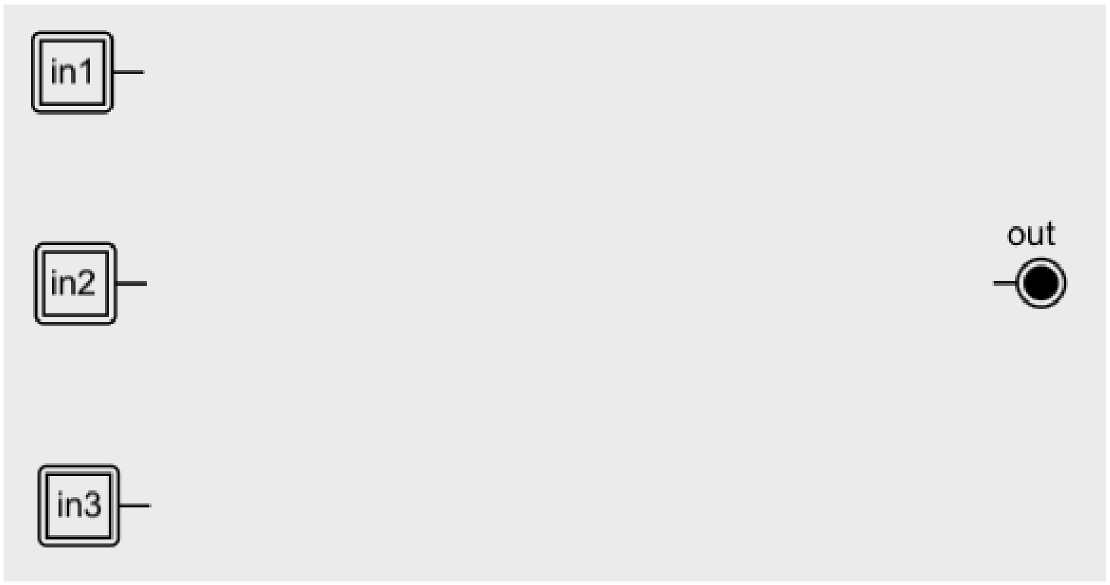
```
in1 = False
```

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

   i.
```
in1 =
in2 =

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

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

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



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

i. `ramble(tia,0)`

ii. `ramble(tia,8)`

```
import turtle
tia = turtle.Turtle()
tia.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-2)
```

(b) For the following code:

```
def v1(vincent, munem):
    if vincent + munem > 0:
        return vincent
    else:
        return -1
```

```
def start():
    panda = 20
    minh = -30
    qiuqun = v1(panda,minh)
    return qiuqun
```

i. What are the formal parameters for `v1()`:

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

iii. What value does `start()` return:

5. Write a function `most_frequent()` that takes a list of 8-digit strings and returns the string that occurs most in the list. If there is a tie for most occurrences, return the first alphabetically. For example:

```
ids = ['12345678','11223344','12312323','12345678']
most_frequent(ids)
```

would return `'12345678'` since it occurs twice, while all other entries once.

Libraries:	
Input:	
Output:	

Design Pattern:

☐ Accumulator ☐ Max/Min ☐ Finding Duplicates ☐ Searching

Principal Mechanisms (select all that apply):

☐ Single Loop ☐ Nested Loop ☐ Conditional (if/else) ☐ Recursion
☐ Indexing/slicing ☐ Dictionary ☐ List Comprehension ☐ Regular Expressions

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

(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 blue (the fraction of blue is above 0.75 and the fraction of green, and the fraction of red are below 0.25).

8. (a) Consider the following MIPS program:

```
ADDI $s0, $zero, 4
ADD $s1, $s0, $s0
ADD $s2, $s1, $s1
SUB $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
ADDI $t0, $zero, 68
ADDI $s2, $zero, 71
SETUP: SB $t0, 0($sp)
ADDI $sp, $sp, 1
ADDI $t0, $t0, 1
BEQ $t0, $s2, DONE
J SETUP
DONE: ADDI $t0, $zero, 0
SB $t0, 0($sp)
ADDI $sp, $sp, -3
ADDI $v0, $zero, 4
ADDI $a0, $sp, 0
syscall
```

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

```
//Derek Bok
#include <iostream>
using namespace std;
int main()
{
  cout << "If you think education";
  cout << endl << "is expens";
  cout << "ive,\nTry ignorance.\n";
  return 0;
}
```

Output:

(b) What is the output:

```
#include <iostream>
using namespace std;
int main()
{
    cout << "Hi" << endl;
    int x = 2;
    while (x > 0) {
        cout <<":)\n";
        x--;
    }
    cout << "Bye"<< endl;
    return 0;
}
```

Output:

(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 ( i % 2 == 0)
                cout<<"0";
            else
                 cout<<"1";
         }
         cout << endl;
     }

      return 0;
}
```

Output:

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

C++ program:

Python program:

```
num = 1
while (num < 100) or (num % 2 == 0):
    num = int(input("Enter large odd #: "))
print("Your number:", num)
```

(b) Write a C++ program that will ask for the time in 24 hour format (e.g. `2034` is 8:34pm) and, prints out "Early" if it is before noon (e.g. `1000`), "Late" if it after 7pm (e.g. `1900`), and otherwise print "Just Right."

A sample run:

```
Enter time:  1345
Good Afternoon
```