

**Answer Key:**

ROW:	SEAT:

MOCK FINAL EXAM  
CSci 127: Introduction to Computer Science  
Hunter College, City University of New York  
December 13, 2022

**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]

(Image from wikipedia commons)

1. (a) Fill in the code below to produce the Output on the right:

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

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

**Output:**  
spring  
summer

**Answer Key:**

```
1 spring_summer = seasons.split(',')[:2]
2 for s in spring_summer:
3     print(s.lower())
```

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

**Output:**  
SUMMER  
WINTER

**Answer Key:**

```
1 summer_winter = seasons.split(',') [1::2]
2 for s in summer_winter:
3     print(s.upper())
```

- (b) Consider the following shell commands:

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

```
i. What is the output for:
$ mkdir projects
$ mv *.py projects
$ cd projects
$ ls
```

**Output:**

**Answer Key:**

```
happy.py hello.py
```

ii. What is the output for:

```
$ pwd
```

**Output:**

**Answer Key:**

```
/usr/student/projects
```

iii. What is the output for:

```
$ cd ..  
$ ls | grep csv
```

**Output:**

**Answer Key:**

```
covid.csv grades.csv
```

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

i. `rgb = (55, 55, 55)`

black       red       white       gray       purple

ii. `rgb = "#AB0000"`

black       red       white       gray       purple

iii. `rgb = (0, 0, 0)`

black       red       white       gray       purple

iv. What is the binary number equivalent of decimal number 45?

Decimal 45 = Binary 

--	--	--	--	--	--

v. What is the Decimal number equivalent to Hexadecimal AC?

Hexadecimal AC = Decimal 

--	--	--

**Answer Key:**

i. `rgb = (55, 55, 55)`

black       red       white       gray       purple

ii. `rgb = "#AB0000"`

black       red       white       gray       purple

iii. `rgb = (0, 0, 0)`

black       red       white       gray       purple

iv. What is the binary number equivalent of Decimal 45?

```

2 | 45
+----
2 | 22  1
+---
2 | 11  0
+---
2 |  5  1
+---
2 |  2  1
+---
2 |  1  0
+---
   |  0  1

```

Decimal 45 = Binary 

1	0	1	1	0	1
---	---	---	---	---	---

v. What is the Decimal number equivalent to Hexadecimal AC?

Hexadecimal AC = Decimal

$$A * 16^1 + C = 10 * 16 + 12 = 172 \begin{array}{|c|c|c|} \hline 1 & 7 & 2 \\ \hline \end{array}$$

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

```
fruits = ['apple', 'banana', 'coconut', 'dragon fruit', 'elderberry']
```

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

i. Answer Key:

```
1 fruits = ['apple', 'banana', 'coconut',
2         'dragon fruit', 'elderberry']
3 for i in range(2):
4     for j in range(len(fruits)):
5         print(fruits[j][0], end=' ')
6     print()
```

Output:

```
a b c d e
a b c d e
```

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

ii.

Answer Key:

```
1 for j in range(len(fruits)-1, -1, -2):
2     print(fruits[j][-1], end=' ')
```

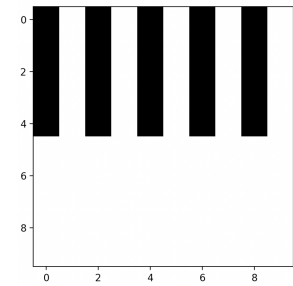
Output:

```
y t e
```

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

iii.

```
1 import numpy as np
2 import matplotlib.pyplot as plt
3
4 img = np.ones((10, 10, 3))
5
6 img[0:5,::2,:] = 0
7 plt.imshow(img)
8 plt.show()
```

**Output:**

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

in1 = True

i. in2 = False

out = not in1 or in2

True

False

**Answer Key:**

out = False

in1 = False

ii. in2 = True

in3 = False

out = not (in1 and not in2) or in3

True

False

**Answer Key:**

out = True

in1 = True

iii. in2 = False

in3 = not in1 or in2

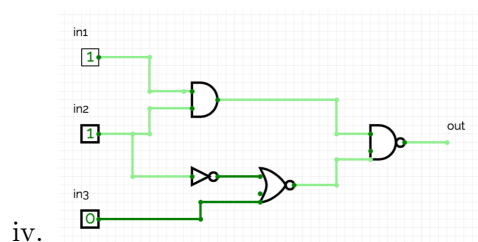
out = not in1 and not in3

True

False

**Answer Key:**

out = False



in1 = True

in2 = True

in3 = False

True

False

**Answer Key:**

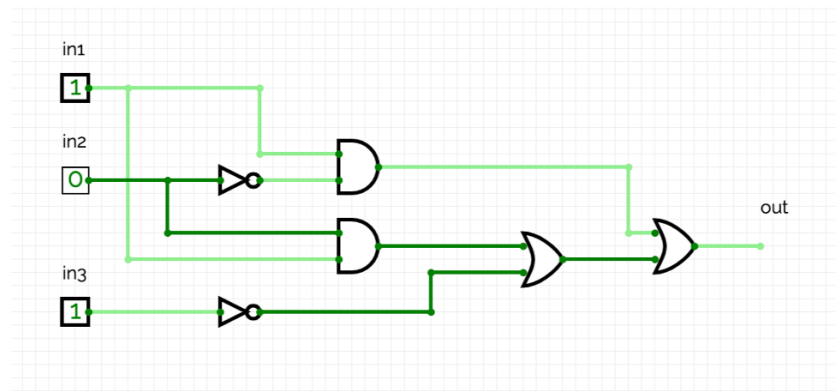
out = False

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

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



Answer Key:



4. Consider the following functions:

```
def count(mylist, target):  
    num_occur = 0  
    for num in mylist:  
        if equal(num, target):  
            num_occur += 1  
  
    return num_occur
```

```
def equal(a, b):  
    return a == b  
  
def main():  
    mylist = [1, 6, 5, 7, 7]  
    print(count(mylist, 6))
```

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

**Answer Key:** a, b

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

**Answer Key:** mylist, 6

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

**Answer Key:** 5

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

i. **Output:**

**Answer Key:**

1

5. Design an algorithm that, given an image, outputs an image that make each pixel its complement. For a pixel with color (r, g, b), its complement color is (1-r, 1-g, 1-b). For example, if a pixel is 100% red, that is, (1, 0, 0), then its complementary color is (0, 1, 1).

**Libraries:**

**Answer Key:** matplotlib.pyplot and numpy

**Input:**

**Answer Key:** The name of the image file

**Output:**

**Answer Key:** The name of output file with complementary color for each pixel. **Design Pattern:**

- Search       Find Min       Find Max       Find All

**Answer Key:**  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()

**Answer Key:**  Search       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.)

**Answer Key:**

- (a) Ask the user for image file name
- (b) Ask the user for output image file name.
- (c) Read the image in a numpy array, call it `img`
- (d) Make a copy of `img` to `img2`.
- (e) Use a nested loop to consider every pixel in `img` looping for rows in outer loop and columns in inner loop
  - i. Get the color of the current pixel of `img`, put return in `r`, `g`, `b`.
  - ii. Set the color of the corresponding pixel of `img2` to be `1-r`, `1-g`, `1-b`.
- (f) Save `img2` to the specified output file name.
- (g) Show `img2` (optional).

An implementation of the above code is as follows (This part is **optional** and will **not** be counted towards grading).

```
1 import matplotlib.pyplot as plt
2 import numpy as np
3
4 inFile = input("Enter input image: ")
5 outFile = input("Enter output image: ")
6
7 img = plt.imread(inFile)
8 img2 = img.copy()
9 height = img.shape[0]
10 width = img.shape[1]
11 for i in range(height):
```

```
12     for j in range(width):
13         r, g, b = img[i,j,:3] #the first three components of the third
14             dimension is r, g, b color channel.
15         img2[i,j,:3] = 1-r,1-g,1-b
16 plt.imsave(outFile, img2)
17 plt.imshow(img2)
18 plt.show()
```

6. Consider the `courses_training.csv` dataset that reports training courses offered in NY state. A snapshot given in the image below:

Organization	Borough	course name	Cost Total	Duration
1st Choice Ca	Brooklyn	Home Health	550	83
A.L.M. Secur	Brooklyn	8 HOUR PRE	500	27
A.L.M. Secur	Brooklyn	16 HOUR OJ	822	20
ACCESS INST	Queens	ESL (Full Pro	4000	750
ACCESS INST	Queens	Home Health	750	83
ACCESS INST	Queens	Medical Assi	6000	600

Fill in the Python program below:

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

```
#Read input data into data frame:
```

```
df = 
```

```
#Calculate hourly_rate by dividing Cost Total by Duration (in hours)
```

```
#Groups the data by Borough to extract data in Queens.
```

```
queens = 
```

```
#Print the minimum, maximum, and average hourly_rate of all training courses in Queens.
```

**Answer Key:**

```
1 import pandas as pd
2
3 df = pd.read_csv("courses_training_simplified.csv")
4 df['hourly_rate'] = df['Cost Total'] / df['Duration']
5 queens = df.groupby('Borough').get_group('Queens')
6 print(queens['hourly_rate'].min())
7 print(queens['hourly_rate'].max())
8 print(queens['hourly_rate'].mean())
```

7. Write a **complete Python program** that prompts the user for the name of a .csv file. Suppose column name of longitude is Longitude and column name for latitude is Latitude and generates an interactive .html map with markers found at each geographical location extracted from the .csv file.

```
#Import the packages for dataframes and for generating html maps
```

```
#Ask user for the name of csv file and store in variable in_file
```

```
#Read the csv file into a dataframe and store it in variable df
```

```
#Create a map and store in variable map
```

```
#Loop through all the rows in the dataframe, create a marker with  
#values found in columns Latitude and Longitude, add marker to the map
```

```
#Save the map to file named map.html
```

### Answer Key:

```
1 #Import the packages for dataframes and for generating html maps  
2 import pandas as pd  
3 import folium  
4  
5 #Ask user for the name of csv file and store in variable in_file  
6 in_file = input("Enter the name of csv file: ")
```



```
7
8 #Read the csv file into a dataframe and store it in variable df
9 df = pd.read_csv(in_file)
10
11 #Create a map and store in variable map
12 map = folium.Map()
13
14 #Loop through all the rows in the dataframe, create a marker with
15 #values found in columns Latitude and Longitude, add marker to the map
16 for index,row in df.iterrows():
17     lt = row['Latitude']
18     lg = row['Longitude']
19     mark = folium.Marker([lt, lg])
20     mark.add_to(map)
21
22 #Save the map to file named map.html
23 map.save(outfile='map.html')
```

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

**Output:**

**Answer Key:**

ace

- (b) Modify the program to print out 6 consecutive letters in decreasing order ('Z' down to 'U'). Shade in the box for each line that needs to be changed and rewrite the instruction below.

- ADDI \$sp, \$sp, -4 # Set up stack
- ADDI \$t0, \$zero, 97 # Set \$t0 at 97 (a)
- ADDI \$s2, \$zero, 3 # Use to test when you reach 3
- SETUP: SB \$t0, 0(\$sp) # Next letter in \$t0
- ADDI \$sp, \$sp, 1 # Increment the stack
- ADDI \$s2, \$s2, -1 # Decrement the counter by 1
- ADDI \$t0, \$t0, 2 # Increment the letter by two
- BEQ \$s2, \$zero, DONE # Jump to DONE if s2 == 0
- J SETUP # Else, jump back to SETUP
- DONE: ADDI \$t0, \$zero, 0 # Null (0) to terminate string
- SB \$t0, 0(\$sp) # Add null to stack
- ADDI \$sp, \$sp, -3 # 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
```

**Answer Key:**

```
1  ADDI $sp, $sp, -7      # Set up stack
2  ADDI $t0, $zero, 90   # Set $t0 at 90 (Z)
3  ADDI $s2, $zero, 6    # Use to test when you reach 6
4  SETUP: SB $t0, 0($sp) # Next letter in $t0
5  ADDI $sp, $sp, 1      # Increment the stack
6  ADDI $s2, $s2, -1     # Decrement the counter by 1
7  ADDI $t0, $t0, -1     # Decrease $t0 by 1
8  BEQ $s2, $zero, DONE  # Jump to DONE if s2 == 0
9  J SETUP                # Else, jump back to SETUP
10 DONE: ADDI $t0, $zero, 0 # Null (0) to terminate string
11 SB $t0, 0($sp)        # Add null to stack
12 ADDI $sp, $sp, -6     # Set up stack to print
13 ADDI $v0, $zero, 4    # 4 is for print string
14 ADDI $a0, $sp, 0      # Set $a0 to stack pointer
15 syscall               # Print to the log
```

9. Fill in the C++ programs below to produce the Output on the right.

```

#include <iostream>
using namespace std;
int main()
{
    for(int i = 7; i <=  ;
 ){
        cout << i+2 << endl;
    }
    return 0;
}

```

**Answer Key:**

```

1  #include <iostream>
2  using namespace std;
3
4  int main()
5  {
(a) 6  for (int i = 7; i <= 13; i += 3)
7      //Warning: do not add ; right after
8      for-head,
9      //or the loop body is empty.
10     //That is,
11     //the following writing is WRONG
12     //for (int i = 7; i <= 13; i += 3) ;
13     //We say, each statement in C++ ends
14     with ;
15     //we do not say, each line in C++
16     ends with ;
17     { //this pair of curly braces can
18     be omitted,
19     //since the loop body has only
20     statement
    cout << i + 2 << endl;
    }
    return 0;
}

```

**Output:**

```

9
12
15

```

```

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

    (b) while(count >=0 && num ) {
        cout << count << " " << num << endl;
        count -= 5;
        num -= 4;
    }
    return 0;
}

```

**Output:**

```

20 10
15 6
10 2

```

**Answer Key:**

```

num >= 2
or
num > 1
#include <iostream>
using namespace std;
int main(){
    (c) for (int i = 9; ; i--){
        cout << "Keep going!" << endl;
    }
    return 0;
}

```

**Output:**

```

Keep going!
Keep going!
Keep going!
Keep going!
Keep going!
Keep going!
Keep going!
Keep going!
Keep going!

```

**Answer Key:**

```

i >= 1
or
i > 0

```

A complete C++ code is as follows.

```
1  #include <iostream>
2  using namespace std;
3
4  int main()
5  {
6      for (int i = 9; i >= 1; i--)
7          { //This pair of curly braces can be omitted
8              //since loop body has only one statement.
9              cout << "Keep going!" << endl;
10         }
11
12     return 0;
13 }
```

If we start  $i$  from 1, and  $i$  is increased by one in each round, until it reaches 9 (included). In each round, print "Keep going!".

Or if we start from 0, then  $i < 9$ , again  $i$  is increased by 1. This is called "shift by 1", that is, when  $i$  starts from 0. then  $i < 9$ , when  $i$  starts from 1, then  $i \leq 9$ . In either case,  $i$  is increased by 1 in each round, in this setting, the loop body runs 9 times – we assume that no other update of  $i$  besides  $i++$  inside the loop body.

```
1  #include <iostream>
2  using namespace std;
3
4  int main()
5  {
6      for (int i = 1; i <= 9; i++)
7          { //This pair of curly braces can be omitted
8              //since loop body has only one statement.
9              cout << "Keep going!" << endl;
10         }
11
12     return 0;
13 }
```

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

```
for i in range(1, 10):  
    for j in range(1, i+1):  
        print(i*j, end=' ')  
    print()
```

```
//include library and namespace
```

```
//main function signature
```

```
{  
    //outer loop line
```

```
{  
    //inner loop line
```

```
//loop body
```

```
}  
//return
```

```
}
```

**Answer Key:**

```
1 #include <iostream>
2 using namespace std;
3
4 int main()
5 {
6     for (int i = 1; i < 10; i++)
7     { // Warning: this pair of curly braces cannot be omitted
8         //since the outer loop body has two or more statements.
9         //One is inner for-loop,
10        //the other is cout << endl; statement.
11        for (int j = 1; j < i+1; j++)
12        { //This pair of curly braces can be omitted,
13            //since the inner loop body has only one statement
14            cout << i * j << " ";
15        }
16        cout << endl;
17    }
18
19    return 0;
20 }
```



- (b) One gallon is 3.78541 liters, it is also equal to 128 oz.  
Write a **complete C++ program** that asks the user for the number of gallons and prints the corresponding number of liters and oz.

```
//include library and namespace
```

```
//main function signature
```

```
{
```

```
  //initialize variables
```

```
  //obtain input
```

```
  //calculate conversions
```

```
  //output conversions
```

```
  //return
```

```
}
```

**Answer Key:**

```
1 #include <iostream>
2 using namespace std;
3
4 int main()
5 {
6     //intialize variables
7     double gallon;
8     double liter;
9     double oz;
10
11     //obtain input
12     cout << "Enter number of gallons: ";
13     cin >> gallon;
14
15     //calculate conversions
16     liter = gallon * 3.78541;
17     oz = gallon * 128;
18
19     //output conversions
20     cout << gallon << " gallons " << liter << " liters" << endl;
21     cout << gallon << " gallons " << oz << " oz" << endl;
22
23     return 0;
24 }
```