| Row: | Seat: |
|---|---|
| | |

# Final Exam F22 V3
# CSci 127: Introduction to Computer Science
# Hunter College, City University of New York

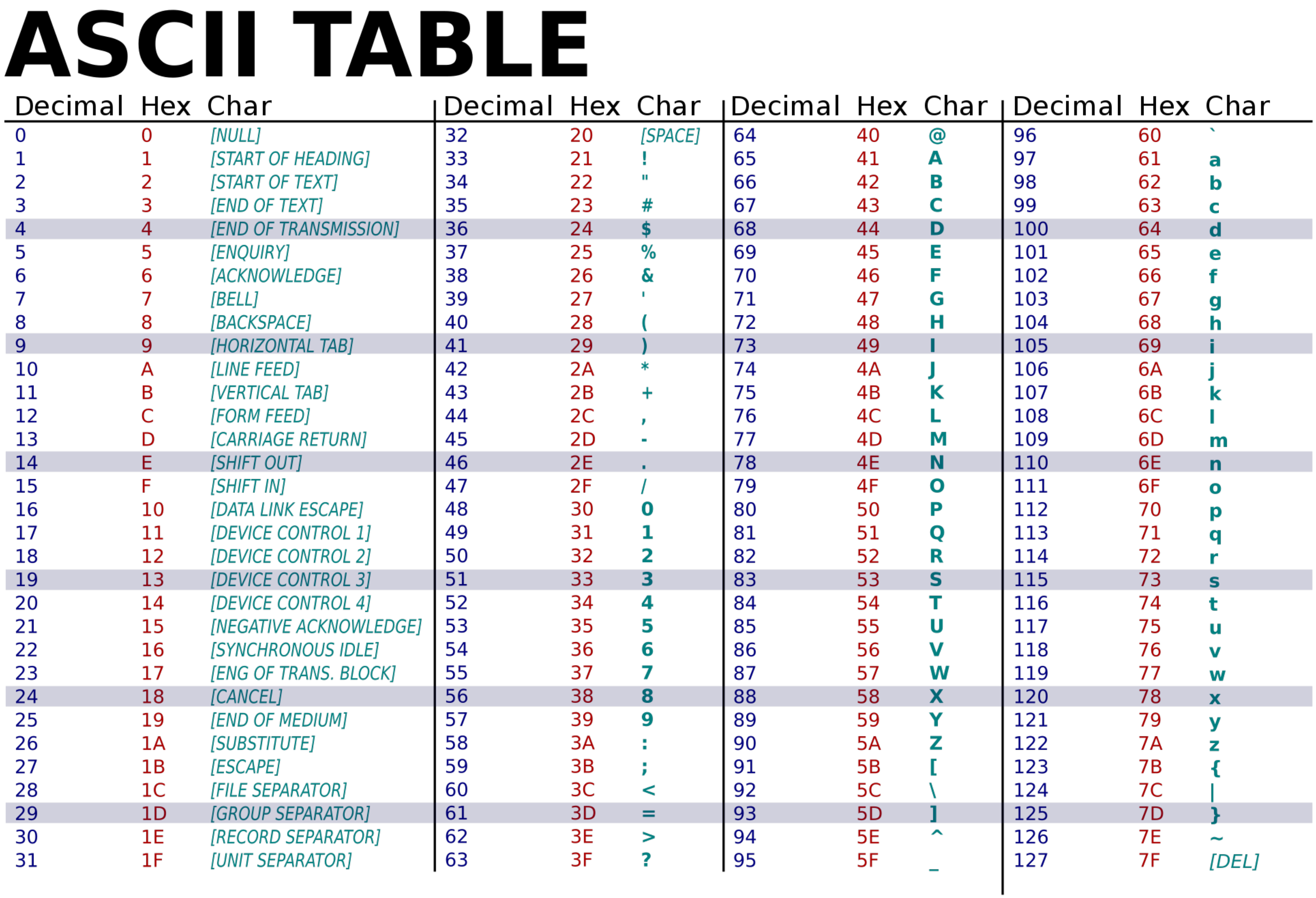
December 19, 2022

## 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:

```
colors='Red-Green-Blue-Yellow-Cyan'
```

i.

```
green = colors[                    ]
print(green)
```

**Output:**

```
Green
```

ii.

```
yellow_green = 
for s in yellow_green:
    print(                    )
```

**Output:**

```
yellow
green
```

(b) Consider the following shell commands:

```
$ pwd
/usr/staff
$ ls
a.out hello.py  p50_growth.cpp p60_binary.cpp
```

i. What is the output for:

```
$ rm a.out
$ mv hello.py p1_hello.py
$ mkdir progs
$ mv *.cpp progs
$ ls
```

**Output:**

ii. What is the output for:

```
$ cd progs
$ pwd
```

**Output:**

iii. What is the output for:

```
$ cd ..
$ ls | grep p | wc -l
```

**Output:**

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

   i. `rgb = (100, 0, 0)`
   □ black □ red □ cyan □ gray □ purple

   ii. `rgb = "#FFFFFF"`
   □ red □ green □ blue □ black □ white

   iii. `What is rgb values for cyan?`
   □ 0, 0, 1 □ 0, 1, 1 □ 1, 0, 0 □ 1, 0, 1 □ 1, 1, 0

   iv. What is the binary number equivalent of decimal number 69?
   `Decimal 69 = Binary` □□□□□□□

   v. What is the Decimal number equivalent to Hexadecimal A6?
   `Hexadecimal A6 = Decimal` □□□

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

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

i.

```
for j in range( □ ):
    print(fruits[□])
```

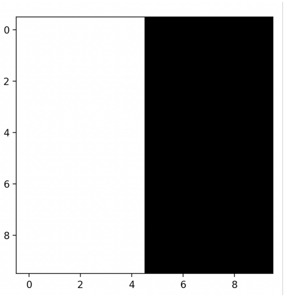
**Output:**

```
coconut
banana
apple
```

ii.

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

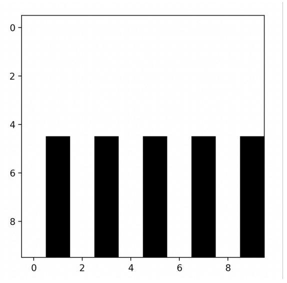
**Output:**



iii.

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

**Output:**



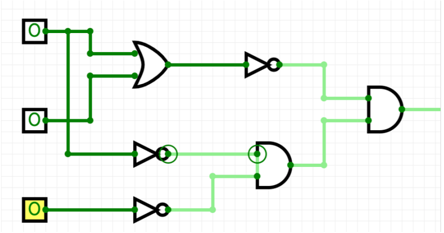
3. (a) What is the value (True/False):

i.
```
in1 = True
in2 = False
out = not (not in1 or in2)
```
□ True □ False

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

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

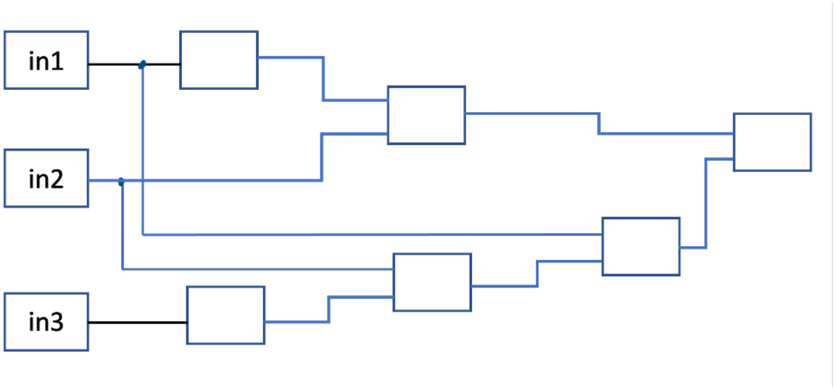
iv.



```
in1 = False
in2 = True
in3 = False
```
□ True □ False

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

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



4. Consider the following functions:

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

    return num_occur
```

```
def leq(s, t):
    return s <= t

def main():
    crr = [21, 32, -55, 91, -26, 72, 1]
    print(count(crr, 32))
```

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

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

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

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

**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) and a threshold. The program outputs the **sum** of all elements in the grid that are larger than or equal to the threshold. For example, suppose the grid has values

```
[[1 2]
 [3 4]]
```

and the given threshold is 3. Then the sum is 3 + 4 = 7.

**Libraries:**

**Input:**

**Output:**

**Design Pattern:**

□ Find Min □ Find Max □ Find All

**Principal Mechanisms (select all that apply):**

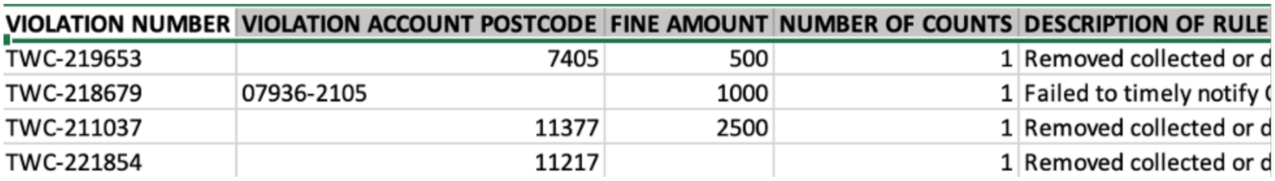
□ Single Loop □ Nested Loop □ Conditional (if/else) statement

□ Indexing / Slicing □ `split()` □ `groupby()`

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

(Assume libraries have already been imported.)

6. Consider the `violations.csv` dataset that reports violations issued by Business Integrity Commission for companies operating in the trade waste industry. A snapshot given in the image below:



| VIOLATION NUMBER | VIOLATION ACCOUNT POSTCODE | FINE AMOUNT | NUMBER OF COUNTS | DESCRIPTION OF RULE |
|---|---|---|---|---|
| TWC-219653 | 7405 | 500 | 1 | Removed collected or d |
| TWC-218679 | 07936-2105 | 1000 | 1 | Failed to timely notify ( |
| TWC-211037 | 11377 | 2500 | 1 | Removed collected or d |
| TWC-221854 | 11217 | | 1 | Removed collected or d |

Assume we write `import pandas as pd` already. Fill in the Python program below:

```
#Read input data into data frame:

df = 
```

```
#Print the min value in column 'NUMBER OF COUNTS'.
```

```
#Groups the data by 'VIOLATION ACCOUNT POSTCODE' to extract data in 10474.

zip10474 = 
```

```
#Print the average of FINE AMOUNT in zip10474.
```

```
#Print the most common (aka top) TEN rules violated.
#Hint: look at 'DESCRIPTION OF RULE' and value_counts method.
```

7. Complete the following code in Python.

Define `diffFreq` function, for strings s1 and s2, char ch, see whether s1 and s2 have different number of occurrences of ch. For example, the return of `diffFreq('abc', 'acd', 'a')` is false since 'a' appears in same frequency in 'abc' and 'acd', but the return of `diffFreq('abc', 'acd', 'b')` is true since 'b' has different number of occurrences in 'abc' and 'acd'.

Define `existDiffFreq` function, for strings s1, s2, and s3, check whether s1 and s2 have different number of occurrences for some letter in s3. For example, existDiffFreq('abcd', 'bcae', 'abc') returns false, since each letter in s3 has the same frequency in s1 and s2, but existDiffFreq('abcd', 'bcae', 'abd') returns true since letter 'd' in s3 has different frequency in s1 and s2.

Hints: once you encounter a letter in s3 that has different number of occurrences in s1 and s2, can you stop and know what `existDiffFreq` function should return immediately? What if after testing every letter in s3, and each one has the same number of occurrences in s1 and s2?

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

**Output:**

(b) Modify the program to print out string ”86420”. Shade in the box for each line that needs to be changed and rewrite the instruction below. Warning: you need to modify from the above code. Need to use j and beq commands.

```
□ ADDI $sp, $sp, -5 # Set up stack

□ ADDI $t0, $zero, 103 # Set $t0 at 103 (’g’)

□ ADDI $s2, $zero, 4 # Use to test when you reach 4

□ 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, 1 # Increase the letter by 1

□ 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, -4 # 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(int i = 3; i <= □ ; □ ){
       cout << i-2  << endl;
    }
    return 0;
}
```

**Output:**

```
1
3
5
7
```

(b)

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

int main()
{
    int size = 4;
    for (int i = 1; i <= size; i++)
    {
        for (int j = 0; j < size - i; j++)
            cout << " ";
        for (int j = 0; j < i; j++)
            cout << "*";
        cout << endl;
    }

    return 0;
}
```

**Output:**

(c)

```
#include <iostream>
using namespace std;
int main(){
    int m = 3;
    int n = 4;

    while ( m + n <= □ ) {

        cout << m << " " << n << endl;

        □ //update m

        n += 3
    }
    return 0;
}
```

**Output:**

```
3 4
1 7
-1 10
-3 13
```

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

```
num = -1
while num < 25 or num > 75:
    num = int(input("Enter an integer in [25, 75]: "))

print("num =", num)
```

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

```
//main function signature
```

```
{
  //initialization

  //loop line

  //loop body
  {

  }
  //print num

  //return

}
```

(b) Write a C++ code. Declare variables for cm and inch. Declare variable for choice. If choice is 1, then enter number of inch, and convert it to cm and print the result out. Otherwise, enter number of cm, and convert it to inch and print the result out.

1 inch = 2.54 cm

1 cm = 1 / 2.54 inch

Some sample input/output is as follows.

```
Enter a choice: 1
Enter number of inch: 5
5 inch = 12.7 cm

Enter a choice: 2
Enter number of cm: 2
2 cm = 0.787402 inch
```

Just finish the code in main function. No need to write include library and main function signature and return statement.

```
//declare variables inch and cm.
```

```
//declare and obtain input for variable choice
```

```
//Write if-statement when choice is 1,
//input inch, convert to cm, and output result.
```

```
//Write else-statement, input cm, convert to inch, and output result.
```

# SCRATCH PAPER

# SCRATCH PAPER