Row:	Seat:

Final Examination, Version 3
CSci 127: Introduction to Computer Science
Hunter College, City University of New York

May 2026

Exam Rules

- Show all your work. Your grade will be based on the work shown.
- You may have pens, pencils and one 8 1/2" x 11" reference sheet filled with notes. No other materials are allowed.
- No phones, computers, tablets, calculators, watches, smart glasses, smart pencils, earpods, or other electronic devices are allowed.
- All electronic devices must be turned off and stored in your bag. If you are not able to turn off the Bluetooth/Wifi on your device, put it in your bag at the front of the room.
- **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:									

1. (a) Assuming the code is run sequentially, what is the output at each step:

i.
```
stops = ["GCT", "51 St", "59 St",\
         "68 St", "77 St", "86 St"]
print(len(stops), "stops")
```
Output:

ii.
```
print("Last:", stops[-1])
```
Output:

iii.
```
trs = ["4,5,7,S","E,F",\
       "4,5,M,N,Q,R","", "", "4,5"]
gct_tr = trs[0]
print(gct_tr)
```
Output:

iv.
```
nums = [t.count(",") for t in trs]
for s,t,n in zip(stops,trs,nums):
    if len(t) > 0:
        print(s,":",n+1,"transfers.")
```
Output:

(b) Consider the following shell commands:

```
$ ls -l
-rw-r--r--    1 stjohn  staff    9901 Jan 19  2025 f24.html
-rw-r--r--    1 stjohn  staff   39765 Dec  9 12:14 f25.html
-rw-r--r--    1 stjohn  staff     972 Jan 28 06:46 index.md
-rw-r--r--    1 stjohn  staff   38332 May 20  2025 s25.html
-rw-r--r--    1 stjohn  staff   40293 Apr  1 16:40 s26.html
```

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

i.
```
$ ls
```
Output:

ii.
```
$ ls -l | grep "Jan" | wc -l
```
Output:

iii.
```
$ echo "Spring:"
$ ls s*.html | wc -l
```
Output:

iv.
```
$ mkdir fall
$ mv f*.html fall
$ ls
```
Output:

2. (a) Check **all that apply**:

   i. What color is the turtle `tom` after this command? `tom.color("#A0A0A0")`
   ☐ black ☐ white ☐ red ☐ blue ☐ None of these

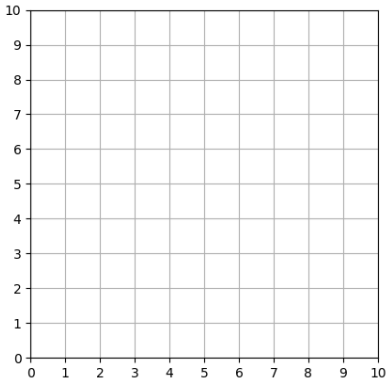
   ii. Select all the binary numbers **larger than 8 decimal**:
   ☐ 0011 ☐ 1101 ☐ 0111 ☐ 0110 ☐ 1110

   iii. Select the hexadecimal numbers **smaller than 160 decimal**:
   ☐ A ☐ AA ☐ B1 ☐ 99 ☐ 9F

   (b) Show the output of this program by shading in the grid:

```
import matplotlib.pyplot as plt
import numpy as np
logo = np.ones( (10,10,3) )
logo[:,:2,:] = 0
logo[-2:,:,:] = 0
logo[:,-2:,:] = 0
plt.imshow(logo, extent= [0,10,0,10])
plt.show()
```

Output:



   (c) For each error below, give the line number and the code that would fix the error.

```
1 import panda as pd
2 dict = {'Boros' : ['B','Bx','M','Q','SI'] 'Pop' : [2.7,1.4,1.6,2.3,0.5]}
3 print(dict.keys()
4 df = pd DataFrame(dict)
5 print( df[['Boros']] )
```

   i.
```
import panda as pd
^^^^^^^^^^^^^^^^^^
ModuleNotFoundError: No module named 'panda'
```
   Line Number: **Code that fixes the error:**

   ii.
```
print(dict.keys()
     ^
SyntaxError: '(' was never closed
```
   Line Number: **Correct code:**

   iii.
```
df = pd DataFrame(dict)
        ^^^^^^^^^
SyntaxError: invalid syntax
```
   Line Number: **Correct code:**

3. (a) What will make the following statement true: **Check all that apply.**

```
in1 and not in2
```

- ☐ Setting `in1 = False` and `in2 = False`.
- ☐ Setting `in1 = False` and `in2 = True`.
- ☐ All values for `in1` and `in2` make the statement true.
- ☐ Setting `in1 = True` and `in2 = False`.
- ☐ Setting `in1 = True` and `in2 = True`.
- ☐ No values for `in1` and `in2` make the statement true.

(b) What is the value of `out`?

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

Choose one:

- ☐ False
- ☐ True
- ☐ Not defined

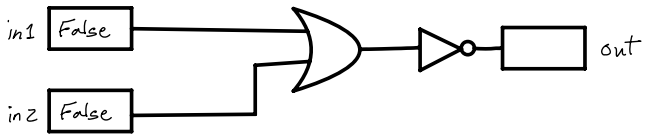
(c) Fill in the values to yield the output:

```
in1 =
in2 =

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

out = True

(d) What is the output of this circuit?

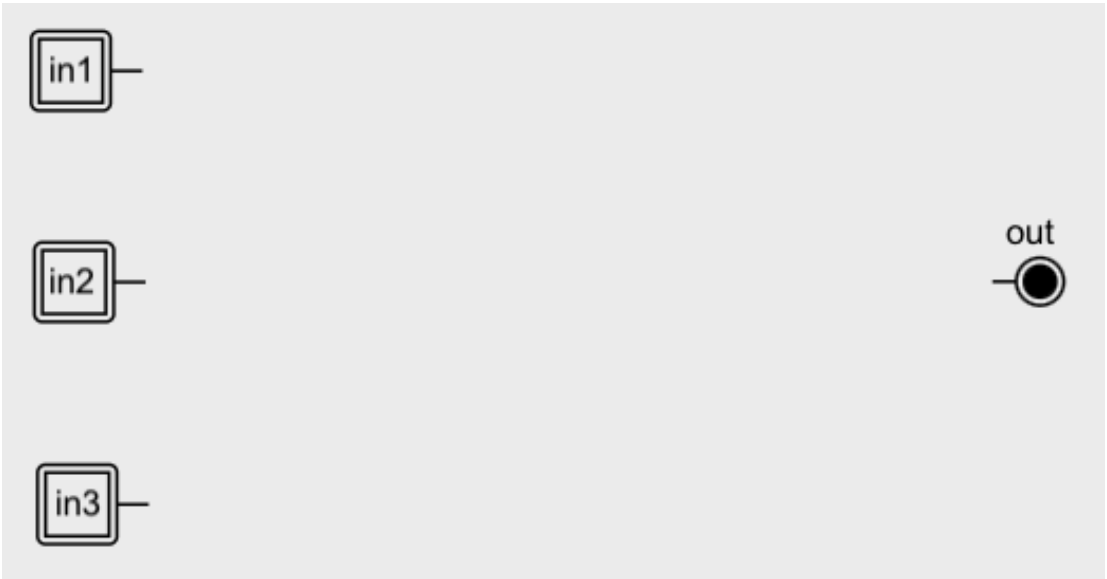


Choose one:

- ☐ False
- ☐ True
- ☐ Not defined

(e) Design a circuit that **exactly implements** the logical expression:

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

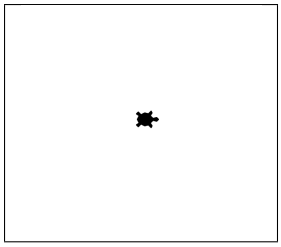


4. (a) Using the turtle and function below, fill in the parameters that yield the output:

```
1  import turtle
2  tim = turtle.Turtle()
3  tim.shape("turtle")
4
5  def ramble(t, side):
6      if side >= 20:
7          for i in range(3):
8              t.forward(side)
9              t.right(360/3)
10         ramble(t, side/2)
11     else:
12         t.stamp()
```

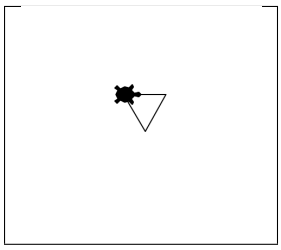
i. `ramble(` _____)

Output:



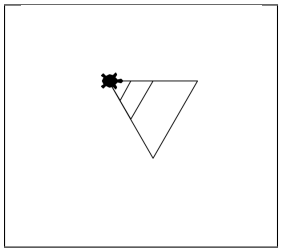
ii. `ramble(` _____)

Output:



iii. `ramble(` _____)

Output:



(b) What are the formal parameters for the function `ramble()`: _____

(c) For which values of `side`, will `ramble(tim,side)` stamp a turtle (check all that apply):

- ☐ `side = 0`
- ☐ `side = 10`
- ☐ `side = 100`
- ☐ `side = 1000`
- ☐ None of the above.

(d) For which values of `side`, will `ramble(tim, side)` **NOT** execute Line 7 (check all that apply):

- ☐ `side = 0`
- ☐ `side = 10`
- ☐ `side = 100`
- ☐ `side = 1000`
- ☐ None of the above.

5. Write a function `search_for()` that takes a list of names and a target returns `True` if the name is in the list and `False` otherwise. For example:

```
names = ['Anany','Anna','Emily','Gordon','Isabelle','Jayden','Rica']
print(search_for(names, 'Anna'))
```

would print `True` since `'Anna'` in the list names.

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 the missing code below to average regions of an image. For example, if you inputted our favorite image, you would see (left to right):

```
#Fill in libraries needed for storing and displaying images:
```

```
def average(region):
    """
    Returns average of red values, of green values, and blue values
    across the inputted region.
    """
```

```
def setRegion(region, r, g, b):
    """
    Takes a region of an image and red, green, and blue values, r, g, b.
    Sets the region so that all points have
    red values of r, green values of g, and blue values of b.
    """
```

```
def quarter(img2, levels):
     hReg = img2.shape[0]//2**levels
     wReg = img2.shape[1]//2**levels
     for i in range(2**levels):
          for j in range(2**levels):
               r,g,b = average(img2[i*hReg:(i+1)*hReg,j*wReg:(j+1)*wReg])
               setRegion(img2[i*hReg:(i+1)*hReg,j*wReg:(j+1)*wReg],r,g,b)
```

7. Write a **complete Python program** that makes a DataFrame to store addresses and saves the DataFrame in a CSV file. Your program should ask the user for:

   - A list of last names,
   - A list of first names,
   - A list of emails, and
   - The name for the output (CSV) file.

   For example, a sample run of your program:

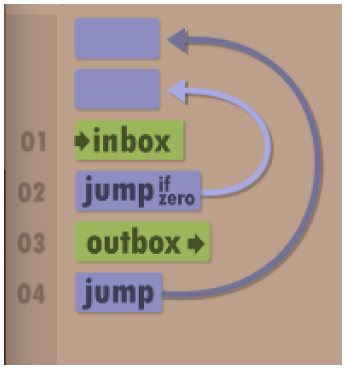
   ```
   Enter last names: Hunter Raab Kirschner Cantor
   Enter first names: Thomas Jennifer Anne Nancy
   Enter emails: th1870@hunter jr2001@hunter ak2023@hunter nc2024@hunter
   Enter file name:  addr.csv
   ```

   would create a DataFrame:

   ```
           Last     First         emails
   0     Hunter    Thomas  th1870@hunter
   1       Raab  Jennifer  jr2001@hunter
   2  Kirschner      Anne  ak2023@hunter
   3     Cantor     Nancy  nc2024@hunter
   ```

   and save the results to `addr.csv`.

8. (a) What does the Human Resource Machine (HRM) code output with the following input:



Output:

Note: if the input is a letter, it is not equal to zero.

(b) Consider the following MIPS program:

```
ADDI $s0, $zero, 1
ADDI $s1, $zero, 2
SUB $s2, $s0, $s1
ADD $s3, $s2, $s2
```

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

$s0 register	$s1 register	$s2 register	$s3 register

(c) Consider the MIPS code:

```
1  ADDI $sp, $sp, -9
2  ADDI $t0, $zero, 58
3  ADDI $t1, $zero, 41
4  ADDI $s2, $zero, 4
5  SETUP: SB $t0, 0($sp)
6  ADDI $sp, $sp, 1
7  SB $t1, 0($sp)
8  ADDI $sp, $sp, 1
9  ADDI $s2, $s2, -1
10 BEQ $s2, $zero, DONE
11 J SETUP
12 DONE: ADDI $t0, $zero, 0
13 SB $t0, 0($sp)
14 ADDI $sp, $sp, -8
15 ADDI $v0, $zero, 4
16 ADDI $a0, $sp, 0
17 syscall
```

i) How many characters are printed?	
ii) What is the first character printed?	
iii) What is the message printed?	
iv) List what you need to change to print half of the message:	

9. (a) Fill in the missing code to yield the output:

```
//Lyrics by Lopez & Lopez
#include <iostream>
using namespace std;
int main()
{
    cout << "It's funny how some ";
    
    cout << "everything seem small" << endl;
    return(0);
}
```

Output:

It's funny how some distance
Makes everything seem small

(b) What is the output:

```
//More Elsa
#include <iostream>
using namespace std;
int main()
{
    int count = 2;
    while (count > 0) {
        cout <<"Let it go, ";
        count--;
    }
    cout << "\nCan't hold it ";
    cout << "back anymore\n";
    return(0);
}
```

Output:

(c) What is the output:

```
#include <iostream>
using namespace std;
int main()
{
    for (int i = 1; i <= 5; i++)
    {
        for (int j = 1; j <= 5; j++)
        {
            if (i % 2 == 0)
                cout << i;
            else
                cout << j;
        }
        cout << "!" << endl;
    }
    return 0;
}
```

Output:

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

C++ program:

```cpp
#include <iostream>
using namespace std;
int main() {
    int times;
    cout << "Enter repetition time: ";
    cin >> times;
    for (int i = 0; i < times; i++) {
        cout<<"Practice makes perfect."<< endl;
    }
    return 0;
}
```

Python program:

(b) Write a C++ program that asks the user for the number of credits completed and prints out number for each subsequent semester until it reaches 120. Each semester, the credits completed increases by 15.

A sample run:

```
Enter starting number of credits: 37
Semester 0:     37
Semester 1:     52
Semester 2:     67
Semester 3:     82
Semester 4:     97
Semester 5:     112
```

SCRATCH PAPER

SCRATCH PAPER

CSCI 127 Reference Sheet, Spring 2026

Turtles: Let `t` be a turtle.

Function	Description
`t=turtle.Turtle()`	Creates turtle `t`.
`t.forward(x)`	Moves `t` forward `x` steps.
`t.backward(x)`	Moves `t` back `x` steps.
`t.left(x)/t.right(x)`	Turns `t` left/right `x` deg.
`t.penup()/t.pendown()`	Lifts `t`'s pen up/down.
`t.stamp()`	Stamps at `t`'s location.
`t.goto(x,y)`	Moves `t` to `(x,y)`.
`t.colormode(255)`	Enable 255 mode.
`t.color(s)/t.color(x,y,z)`	Changes `t`'s color.

String Methods: Let `s` be a string.

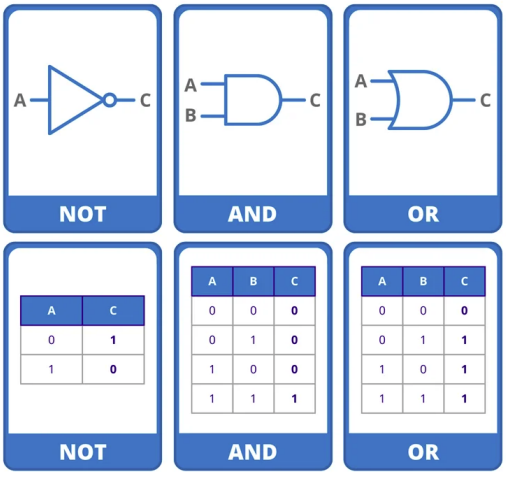
Function	Description
`len(s)`	Returns the length of `s`.
`s.lower()`	Returns `s` as lower case character(s).
`s.upper()`	Returns `s` as upper case character(s).
`s.count(t)`	Returns count of t in s.
`s.find(t)`	Returns index of t in s (-1 not found).
`s.split(d)`	Splits `s` into list of strings on `d`.
`s.join(lst)`	Joins `lst` into a string, by `s`.
`s[i:j]`	Substring (slice) of s: from i to j-1.
`ord(c)`	Returns Unicode/ASCII of `c`.
`chr(i)`	Returns character of `i`.

Containers: Lists, Ranges & Dictionaries.

Function	Description
`l = []`	Creates an empty list.
`l = [a,b,c]`	List with 3 elements.
`l.append(elt)`	Adds `elt` to end of list.
`l[i]`	Access element at index `i`.
`range(start,stop,step)`	Range object from `start` to `stop-1`, by `step`.
`zip(l1,l2)`	Combines `l1` & `l2` pairwise.
`[x*x for x in l1]`	List of `l1`'s elements squared. (using list comprehension).
`d = {}`	Creates an empty dictionary.
`d = {k1:v1,k2:v2}`	Dictionary of key/value pairs.
`d[k] = v`	Adds `k:v` to dictionary.
`d[k]`	Access value at key `k`.
`k in d`	Checks if key is in dictionary.
`d.keys() / d.values()`	Returns keys/values of `d`.

Functions:

Function	Description
def fname(x,y):	Defines function, `fname`, with
`command1`	(formal) input parameters, `x` and `y`.
`command2...`	Body of function indented.
`return(v)`	Returns value `v`.
`c = fname(a,b)`	Calls/invokes `fname` with (actual) parameters `a` & `b`, returns to `c`.



(from truthtablegen.com)

Control Structures:

Function	Description
`for i in range(0,100,5):`	Basic for-loop.
`for c in str_var:`	Looping through a string.
`if x==0:... elif x>0:... else:...`	Basic if-statement.

numpy: Let `np` be the `numpy` package.

Function	Description
`arr_z = np.zeros((10,20,3))`	Sets up array for 10x20 black image.
`arr_1 = np.ones((10,20,3))`	Sets up array for 10x20 white image.
`arr[start:stop:step]`	Slice from `start` to `stop-1` by `step`.
`arr = plt.imread('image.png')`	Read in an image.
`plt.imshow(arr)`	Show `arr` as image.
`plt.show()`	
`plt.imsave('image.png', arr)`	Save an array to file.

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

Function	Description
`pd.read_csv(fn)`	Returns a DataFrame with file `fn`.
`pd.DataFrame(d)`	Returns DataFrame of dictionary `d`.
`df.to_csv(file_name)`	Writes `df` to `file_name`.
`df[col]`	Returns `col` column as a Series.
`df[[c1,c2]]`	Returns DataFrame with `c1` & `c2`.
`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`.
`df.dropna(subset=[c1,c2])`	Drops rows with missing c1 or c2 values.

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 -l / ls|grep pat`	Uses pipes to count # of files/match `pat`

C++:

Function	Description
`#include <iostream>`	Includes library with `cin/cout`.
`using namespace std;`	Use standard names w/o `std::`.
`int main() {...}`	Function definition.
`int x;`	Declares variable `x` to be an integer.
`float y;`	Declares variable `y` to be a float.
`cin >> x;`	Reads input into `x`.
`cout << x;`	Prints `x`.
`for (i=0; i<10; i++){...}`	Basic for-loop.
`while (logicalExpression){...}`	Basic while-loop.
`return(v);`	Returns value `v`.

(From wikipedia commons)

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]