Mock Final Exam

CSci 127: Introduction to Computer Science
Hunter College, City University of New York

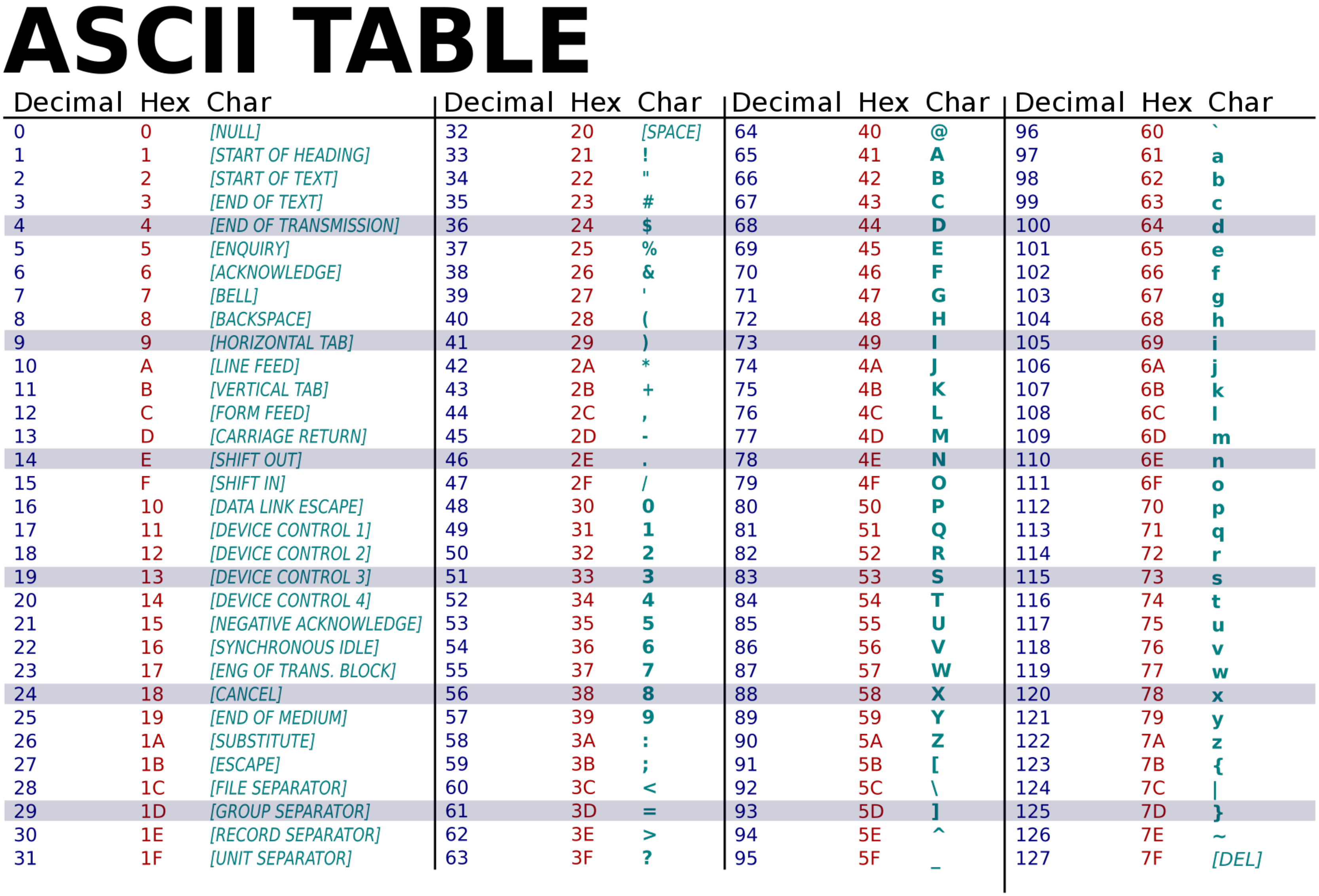
10 December 2019

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, 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) What will the following Python code print:

```
pioneers="Asimov%Isaac#Shelley%Mary#Gibson%William"
```

i.

```
num = pioneers.count('%')
num = num + pioneers.count('#') + 2
print(pioneers[len(pioneers)-num:].lower())
```

Output:

ii.

```
names = pioneers.split('#')
m = names[1]
print(m[-4]+'. '+m[:7])
```

Output:

iii.

```
for n in names:
    print(n.split('%')[0][0])
```

Output:

(b) Consider the following shell commands:

```
$ pwd
/Users/login/temp
$ ls
csBridge.png Elevations.csv  p25.py  p27.py
```

i. What is the output for:

```
$ mkdir hwk
$ mv *.p* hwk
$ ls
```

Output:

ii. What is the output for:

```
$ cd hwk
$ ls | grep ^p | wc -l
```

Output:

iii. What is the output for:

```
$ pwd
$ cd ../
$ pwd
```

Output:

2. (a) Consider the code:

```
import turtle
thomasH = turtle.Turtle()
```

i. After the command: `thomasH.color("#1B1B1B")`, what color is `thomasH`?
□ black □ red □ white □ gray □ teal

ii. After the command: `thomasH.color("#00AAAA")`, what color is `thomasH`?
□ black □ red □ white □ gray □ teal

iii. Fill in the code below to change `thomasH` to be the color white:

```
thomasH.color("# □ □ □ □ □ □ ")
```

iv. Fill in the code below to change `thomasH` to be the brightest purple:

```
thomasH.color("# □ □ □ □ □ □ ")
```

(b) Fill in the code to produce the output on the right:

i.

```
for i in range( □ ):
    print(i, end=" ")
```

Output:

```
0 1 2 3 4 5 6 7 8 9
```

ii.

```
for j in range( □ , □ , □ ):
    print(i, end=" ")
```

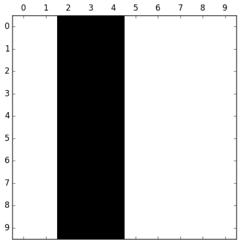
Output:

```
-2 0 2 4 6
```

iii.

```
import numpy as np
import matplotlib.pyplot as plt
im = np.ones( (10,10,3) )
im[:,□:5,:]  = 0
plt.imshow(im)
plt.show()
```

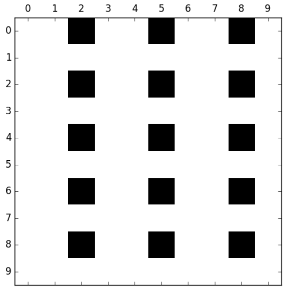
Output:



iv.

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

Output:



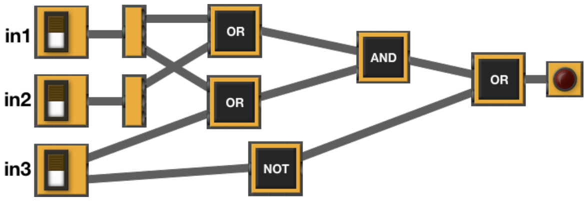
3. (a) What is the value (True/False):

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

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

iii.
```
in1 = True
in2 = True or not in1
in3 = in1 or in2
out = in1 and not in3
```
☐ True ☐ False

iv.



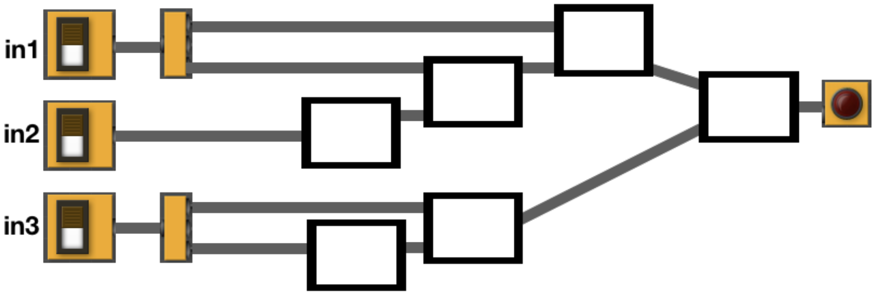
```
in1 = True
in2 = False
in3 = False
```
☐ True ☐ False

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

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

(c) Fill in the circuit that implements the logical expression:

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



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

```
1:  import turtle
2:  tess = turtle.Turtle()
3:  tess.shape('turtle')

4:  def ramble(t, len):
5:      if len <= 10:
6:          t.stamp()
7:      elif len%2 == 0:
8:          t.left(90)
9:          t.forward(len)
10:         ramble(t, len//2)
11:     else:
12:         t.right(90)
13:         t.forward(len)
14:         ramble(t, len//2)
```

i. `ramble(tess,8)`

ii. `ramble(tess,180)`

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

(c) If you call `ramble(tess,8)`, which branches of the function are tested (check all that apply):

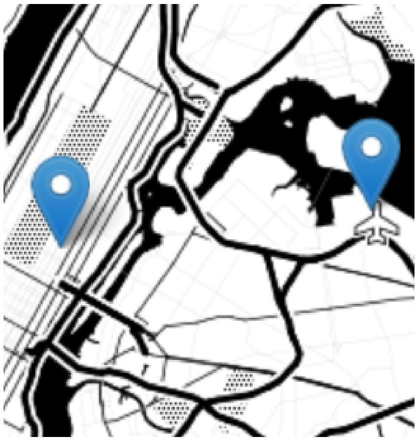
- □ The block of code at Line 6.
- □ The block of code at Lines 8-10.
- □ The block of code at Lines 12-14.
- □ None of these blocks of code (lines 6, 8-10, 12-14) are visited from this invocation (call).

(d) If you call `ramble(tess, 180)`, which branches of the function are tested:

- □ The block of code at Line 6.
- □ The block of code at Lines 8-10.
- □ The block of code at Lines 12-14.
- □ None of these blocks of code (lines 6, 8-10, 12-14) are visited from this invocation (call).

5. Consider the driving times from Hunter College to LaGuardia Airport using the different routes which factors in delays due to traffic (x cars) already en route:

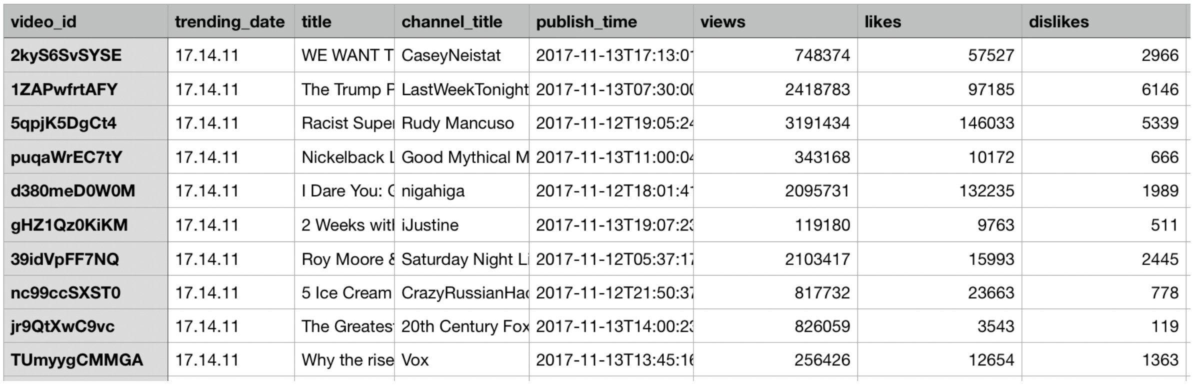
- $\mathrm{T}_{RFK}(x) = 14 + \frac{x}{10{,}000}$, the time, in minutes, for the Triborough/RFK bridge route.
- $\mathrm{T}_{KQB}(x) = 18 + \frac{x}{5{,}000}$, the time, in minutes, for the Koch Queensboro bridge route.
- $\mathrm{T}_{Tun}(x) = 16 + \frac{x}{1{,}000}$, the time, in minutes, for the Queens Midtown Tunnel route.



Design an algorithm that, given the current status of traffic at any given moment, suggests the best route.

- **Input:**
- **Output:**
- **Process (as a list of steps):**

6. Given the YouTube dataset of top-trending videos in 2017, a snapshot given in the image below:



video_id	trending_date	title	channel_title	publish_time	views	likes	dislikes
2kyS6SvSYSE	17.14.11	WE WANT T	CaseyNeistat	2017-11-13T17:13:0	748374	57527	2966
1ZAPwfrtAFY	17.14.11	The Trump F	LastWeekTonight	2017-11-13T07:30:0	2418783	97185	6146
5qpjK5DgCt4	17.14.11	Racist Supe	Rudy Mancuso	2017-11-12T19:05:2	3191434	146033	5339
puqaWrEC7tY	17.14.11	Nickelback L	Good Mythical M	2017-11-13T11:00:0	343168	10172	666
d380meD0W0M	17.14.11	I Dare You:	nigahiga	2017-11-12T18:01:4	2095731	132235	1989
gHZ1Qz0KiKM	17.14.11	2 Weeks wit	iJustine	2017-11-13T19:07:2	119180	9763	511
39idVpFF7NQ	17.14.11	Roy Moore	Saturday Night L	2017-11-12T05:37:1	2103417	15993	2445
nc99ccSXST0	17.14.11	5 Ice Cream	CrazyRussianHa	2017-11-12T21:50:3	817732	23663	778
jr9QtXwC9vc	17.14.11	The Greates	20th Century Fox	2017-11-13T14:00:2	826059	3543	119
TUmyygCMMGA	17.14.11	Why the rise	Vox	2017-11-13T13:45:1	256426	12654	1363

Fill in the Python program below:

```
#P6, Mock: extracts data about indifferent views and about videos
#with highest number of likes
#Import the libraries for data frames and plotting data:



#Prompt user for input file name:

csvFile =

#Read input data into data frame:

youtube =

#Calculate a new column for the number of indifferent views
#(i.e. those views that did not like nor dislike)



#Print the maximum number of indifferent views on a video



#Group videos by channel to find out the maximum number of likes on each channel

channelLikes = youtube.groupby(["            " ])["            " ].max()

#Print the top 5 channels with largest number of likes

print(channelLikes[    :    ] )
```

7. Write a **complete Python program** that prompts the user for the name of an .png (image) file and prints the fraction of pixels that are grayscale, or a shade of gray. Recall that a pixel is a shade of gray if the red, green, and blue values are all equal.

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

   Output:

   (b) Modify the program to print out 10 consecutive letters starting with ’K’. Shade in the box for each line that needs to be changed and rewrite the instruction in the space below.

```
□ ADDI $sp, $sp, -6        # Set up stack

□ ADDI $t0, $zero, 75      # Start $t0 at 75 (K)

□ ADDI $s2, $zero, 83      # Use to test when you reach 83 (S)

□ SETUP: SB $t0, 0($sp)    # Next letter in $t0

□ ADDI $sp, $sp, 1         # Increment the stack

□ ADDI $t0, $t0, 2         # Increase the letter by 2

□ BEQ $t0, $s2, DONE       # Jump to done if $t0 == 83

□ J SETUP                  # If not, jump back to SETUP for loop

□ DONE: ADDI $t0, $zero, 0 # Null (0) to terminate string

□ SB $t0, 0($sp)           # Add null to stack

□ ADDI $sp, $sp, -5        # Set up stack to print

□ ADDI $v0, $zero, 4       # 4 is for print string

□ ADDI $a0, $sp, 0         # Set $a0 to stack pointer for printing

□ syscall                  # Print to the log
```

9. What is the output of the following C++ programs?

(a)

```
//Quote by Mary Shelley
#include <iostream>
using namespace std;
int main()
{
    cout<<"Invention,\nit must be ";
    cout<<"humbly admitted,\ndoes not ";
    cout<<"consist in ";
    cout<<"creating"<<endl<<"out of ";
    cout<<"void,\nbut out of chaos. ";
    cout<<"M.S.";
    return 0;
}
```

Output:

(b)

```
#include <iostream>
using namespace std;
int main()
{
    double tot = 0;
    cout <<"Please enter amount\n";
    cin >> tot;
    while (tot > 100) {
        tot = tot - (tot * 0.5);
        cout << tot << endl;
    }
    return 0;
}
```

Input: 400; Output:

(c)

```
#include <iostream>
using namespace std;
int main(){
    int i, j;
    for (i = 1; i < 6; i++){
        for (j = 1; j < 6; j++){
            if(j % 2 == 1)
                cout << i;
            else
                cout << j;
        }
        cout << endl;
    }
    return 0;
}
```

Output:

10. Write a **complete C++ program** that repeatedly asks the user for their score on a programming assignment until the entered score is a negative number. The program then **prints the average programming assignment score**. The negative number simply indicates that the user has finished entering scores and it is not included in the average.