CSCI.UA.0002
Midterm Exam #2 – PRACTICE
Introduction to Computer Programming (Python)

Name: _____

NetID / Email: _____

Test Version: LINCOLN

General notes:

- When writing programs, comment your code as necessary.
- You will lose a small # of points for syntax errors so double-check your work to ensure that it is syntactically correct.

1. Trace the output of the following code samples

```
for number in [100, 130, 10]:
     print(“Number is: “, number)
```

```
for number in range(100, -20, 20):
     print("Number is: ",number)
```

```
for number in range(5, 10, 2):
     print(“Number is: “,number)
```

```
for x in range(2):
     for y in range(3):
          print (x,y)
```

```
for x in range(2):
     y = 2
     while y > 0:
          print (x, y)
          y -= 1
```

2. Given the following String:

```
phrase = "dora the explorer"
```

Identify the value of each of the following items:

```
phrase[0]

phrase[5]

phrase[5:8]

phrase[:8]

phrase[::3]

phrase[5:]

phrase[0:4].isalnum()

phrase[0:5].isalnum()

phrase[-1]

phrase.replace("dora", "python")

phrase.find("dora")

phrase.isalpha()
```

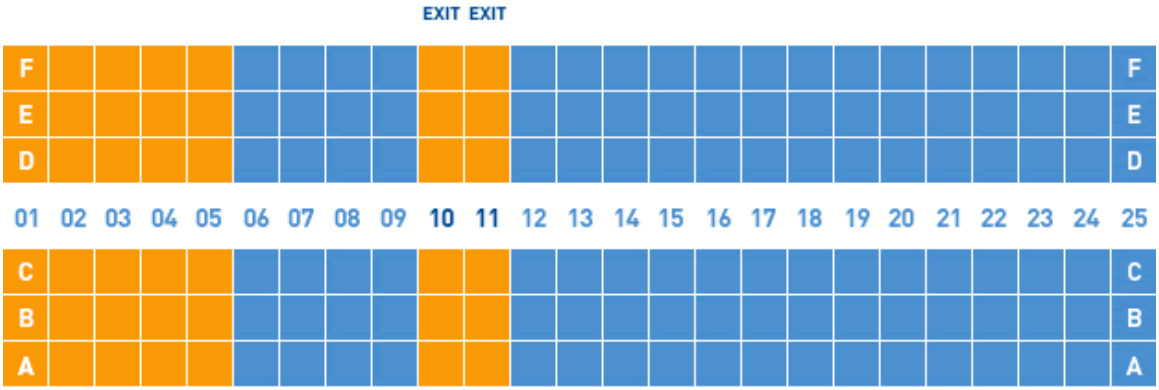
3. Trace the output of the following programs:

```
def b(y):
    print ("$:", y)
    y += 1
    print ("^:", y)

y = 0
print ("(:", y)
b(y)
print ("):", y)
```

```
def m(i):
    i = i ** 2
    return i

def p(i):
    i = i // 2
    return i

print (m(2), p(2))
```

```
def one(a):
    print ("a. one:", a)
    a = two(a)
    print ("b. one:", a)

def two(a):
    print ("a. two:", a)
    a += 1
    print ("b. two:", a)
    return a

one(5)
```

4. An airplane has a total of 25 rows. Each row has 6 seats labeled A, B, C, D, E and F. Seats A and F are "window" seats, seats B and E are "middle" seats and seats C and D are "aisle" seats.



You are a programmer working for an airline that prices seats on its planes using the following algorithm:

- Seats in the "exit row" (rows 10 & 11) cost $200
- Seats in the first 5 rows of the airplane (rows 1-5) cost $500
- All other seats cost $150
- All middle seats (B and E) in any section are discounted by $50 due to the fact that they are awful.

Write a ***FUNCTION*** that accepts two arguments (the row # and a desired seat). Compute the cost of that seat using the pricing chart above and ***RETURN*** the result. Note that your function should be able to handle invalid seats (i.e. row 40 seat Q doesn't exist) – in this case you should return an "invalid seat" string. Comment your function using IPO notation. Here's a sample program that uses your function:

```
print ("20 A: ", seat_cost(20, 'A'))      # 20 A: 150
print ("25 B: ", seat_cost(25, 'B'))      # 25 B: 100
print ("10 A: ", seat_cost(10, 'A'))      # 10 A: 200
print ("11 B: ", seat_cost(11, 'B'))      # 11 B: 150
print ("1 A: ",  seat_cost(1,  'A'))      #  1 A: 500
print ("2 B: ",  seat_cost(2,  'B'))      #  2 B: 450
print ("25 Q: ", seat_cost(25, 'Q'))      # 25 Q: invalid seat
print ("99 A: ", seat_cost(99, 'A'))      # 99 A: invalid seat
print ("85 X: ", seat_cost(85, 'X'))      # 85 X: invalid seat
```

Note: you are not writing a full program for this question -you will do that in the next question. Just write the function as specified.

5. Write a program that prompts the user to enter in a row # and seat #. Next, check the cost of that seat using the function you wrote for the previous question. If the seat is valid you should add the cost of the seat to the user's cart and provide them with a running total of their bill. Here's a sample running of your program:

```
Enter a row number (1-25): 20
Enter a seat (A,B,C,D,E or F): A
This seat will cost 150
Your total is currently 150
Would you like to buy another seat? yes

Enter a row number (1-25): 25
Enter a seat (A,B,C,D,E or F): B
This seat will cost 100
Your total is currently 300

Would you like to buy another seat? yes
Enter a row number (1-25): 1
Enter a seat (A,B,C,D,E or F): Q
Invalid seat
Enter a row number (1-25): 1
Enter a seat (A,B,C,D,E or F): A
This seat will cost 500
Your total is currently 800
Would you like to buy another seat? no

The total cost for your flight will be: 800
```

6. A DNA strand consists of a sequence of smaller molecules called "nucleotides" – there are four main nucleotides represented by the following four characters:

- C = Cytosine
- T = Thymine
- A = Adenine
- G = Guanine

Biologists often need to analyze a sequence of DNA to determine what nucleotides are present in order to learn more about the genes that are represented in that strand of DNA.

Your task for this problem is to write a program that analyzes a strand of DNA. Here's an example strand:

```
DNA_sequence = '_CTAG ttt gABBa!'
```

Note that the sequence above contains valid nucleotides (C, c, T, t, A, a, G, g) as well as a number of invalid nucleotides (spaces, invalid characters, etc.) You will need to "clean up" the sequence so that only the "good" characters remain. For example, the sequence above would "clean up" to the following sequence:

```
good_DNA_sequence = 'CTAGTTTGAA'
```

Once you have done this you should print the following information:

1. The cleaned up DNA sequence (convert valid lowercase nucleotides to uppercase, remove invalid characters)
2. Total number of valid nucleotides found
3. Total number of invalid characters found

Here's a sample running of the program:

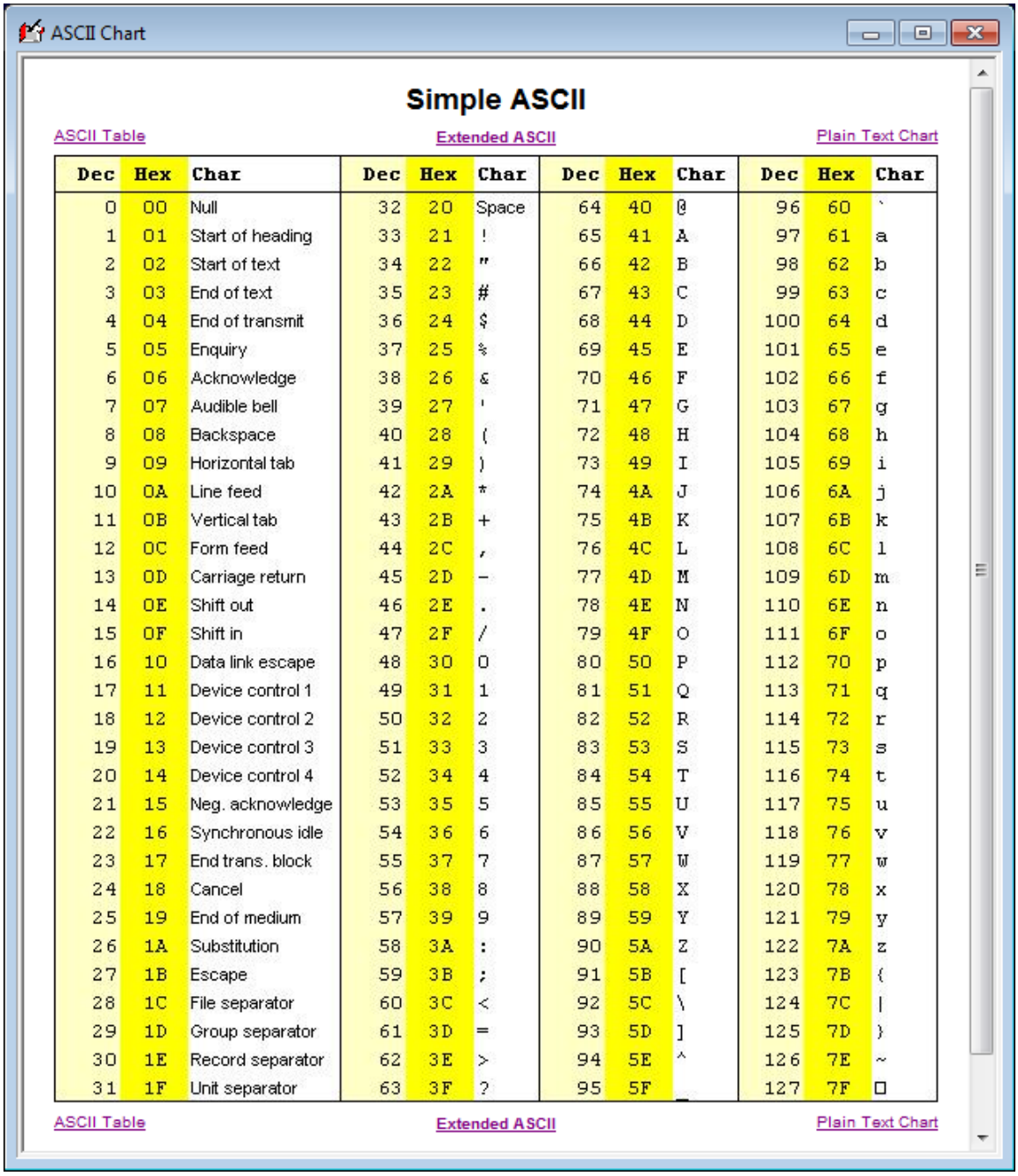
```
Original DNA Sequence: _CTAG ttt gABBa!
Good DNA Sequence: CTAGTTTGAA
Valid nucleotides found: 10
Invalid characters found: 6
```

7. Write a program that lets a student in our class input their scores for 10 homework assignments. You can assume that all homework assignments are worth 20 points. Ensure that the numbers entered are greater than 0, but numbers greater than 20 are OK (we like extra credit!). You can assume that the user will enter an integer or a float. If they input a negative number (or zero) you should continually prompt them to enter a valid number until they do so. Then use their input to generate a summary report that includes the total points earned, total points possible and average score. Below is a sample running of this program. Don't worry about formatting your numbers for this problem.

```
Score 1: 19
Score 2: 18
Score 3: -1

Sorry, -1 is not a valid score.  Please try again.

Score 3: 17
Score 4: 19
Score 5: 20
Score 6: 10
Score 7: 15
Score 8: 14
Score 9: 18
Score 10: 19

Total points earned: 169.0
Total points possible: 200
Average homework score: 16.9 / 20.0
```

Python Command Index

Core Language Elements and Functions	**Module Functions**
and	random.randint
chr	
def	**String Testing Methods**
del	isalpha()
elif	isdigit()
else	islower()
except	isupper()
float	isspace()
for	isalnum()
format	find()
global	
if	**String Modification Methods**
import	rstrip()
in	lstrip()
input	lower()
int	upper()
max	capitalize()
min	title()
not	swapcase()
open	replace()
or	
ord	
print	
range	
return	
str	
str.lower	
str.upper	
try	
while	



Simple ASCII

ASCII Table | Extended ASCII | Plain Text Chart

Dec	Hex	Char	Dec	Hex	Char	Dec	Hex	Char	Dec	Hex	Char
0	00	Null	32	20	Space	64	40	@	96	60	`
1	01	Start of heading	33	21	!	65	41	A	97	61	a
2	02	Start of text	34	22	"	66	42	B	98	62	b
3	03	End of text	35	23	#	67	43	C	99	63	c
4	04	End of transmit	36	24	$	68	44	D	100	64	d
5	05	Enquiry	37	25	%	69	45	E	101	65	e
6	06	Acknowledge	38	26	&	70	46	F	102	66	f
7	07	Audible bell	39	27	'	71	47	G	103	67	g
8	08	Backspace	40	28	(	72	48	H	104	68	h
9	09	Horizontal tab	41	29	)	73	49	I	105	69	i
10	0A	Line feed	42	2A	*	74	4A	J	106	6A	j
11	0B	Vertical tab	43	2B	+	75	4B	K	107	6B	k
12	0C	Form feed	44	2C	,	76	4C	L	108	6C	l
13	0D	Carriage return	45	2D	-	77	4D	M	109	6D	m
14	0E	Shift out	46	2E	.	78	4E	N	110	6E	n
15	0F	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	Neg. 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	End 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	Substitution	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	□

ASCII Table | Extended ASCII | Plain Text Chart