Programming Workshop 4 SOLUTIONS

Module 7 + 8

1. Given the following variables, what will print when the single lines of code below are executed? If a line crashes provide a brief description of the error.

```
a_string = "There is less than one week left of school! You can do it!"
a_list = ["apples", True, "2", 4, 5.0]
```

```
print ( "impossible" in a_string )          False

print ( a_string.isalnum() )                False

print ( a_string[:5]  )                     There

print ( a_string[-1:-4:-1] )                !ti

print ( a_string[44::] )                    an do it!

print ( a_list[2].isdigit() )               True

print ( a_list[0] )                         apples

print ( a_list.index(True) )                1

print ( a_string[a_list[3]].upper() )       E

print ( len(a_list) )                       5
```

2. **True or False.**

Statement	True / False

Program #1
`mylist = []`
`mylist.append("Pokemon")`
`print(mylist)`
 	
Program #2
`mylist = []`
`mylist += "Pokemon"`
`print(mylist)`
 	
Program #1 and Program #2 produce the same output.
	`False`
	
	`['Pokemon']`
	
	`['P', 'o', 'k', 'e', 'm', 'o', 'n']`

Program #1
`mylist = ["A", "B", "C"]`
`mylist.remove("C")`
`print(mylist)`
 	
Program #2
`mylist = ["A", "B", "C"]`
`mylist.remove(mylist[len(mylist)])`
`print(mylist)`
 	
Program #1 and Program #2 produce the same output.
	`False`
	
	`['A', 'B']`
	
	`IndexError: list index out of range`

`scrambled =["z","p","i","y","e","k","u","n","g","s"]`
`secretnumber = 80004070260090`
 	
`for c in str(secretnumber):`
`  if c != "0":`
`  print (scrambled[int(c)],end ="")`
 	
This program will output the word `"pies"`
	`False`
	
	`genius`

`print("Pokemon"[::])` is a valid line of code.	`True`
The `find()` method works for Strings and Lists.	`False`
The `index()` method returns -1 when a value is not found.	`False`

3. **Character Analysis**

Ask the user for a sentence. Then write a program that analyzes the sentence and determines the following:

- The number of uppercase letters in the string
- The number of lowercase letters in the string
- The number of digits in the string
- The number of space characters in the string

Bonus: Keep asking the user for a sentence until they enter the word “no”. For each new sentence, report the statistics above.

```
mystring = input("Enter a sentence: ")

upper = 0
lower = 0
digits = 0
spaces = 0

for c in mystring:
    #print(c)
    if c.isupper():
        upper += 1
    elif c.isspace():
        spaces += 1
    elif c.islower():
        lower += 1
    elif c.isdigit():
        digits += 1

print("Upper:", upper)
print("Lower:", lower)
print("Space:", spaces)
print("Digit:", digits)
```

OR

```
mystring = input("Enter a sentence: ")

upper = 0
lower = 0
digits = 0
spaces = 0

for c in mystring:
    #print(c)
    if ord(c) >= 65 and ord(c) <= 90:
        upper += 1
    elif c.isspace():
        spaces += 1
    elif ord(c) >= 97 and ord(c) <= 122:
        lower += 1
    elif ord(c) >= 48 and ord(c) <= 57:
        digits += 1

print("Upper:", upper)
print("Lower:", lower)
print("Space:", spaces)
print("Digit:", digits)
```

Bonus:

```
mystring = input("Enter a sentence: ")

while mystring != "no":
    upper = 0
    lower = 0
    digits = 0
    spaces = 0

    for c in mystring:
        #print(c)
        if c.isupper():
            upper += 1
        elif c.isspace():
            spaces += 1
        elif c.islower():
            lower += 1
        elif c.isdigit():
            digits += 1

    print("Analyzing '"+ mystring + "':")
    print("Upper:", upper)
    print("Lower:", lower)
    print("Space:", spaces)
    print("Digit:", digits)

    print()
    mystring = input("Enter a sentence: ")


print("All done!")
```

4. **Lottery Number**

Write a program that generates a random 5 digit lottery number. Numbers must be between 1 and 60. No duplicate numbers are permitted. Sort the numbers in ascending order and print them out to the user.

Sample output:

```
Your numbers are: 3 8 17 34 43
```

```
import random

# create empty list to store numbers
numbers = []

# create 5 numbers
for i in range(5):
    pick = random.randint(1,60)

    # keep repeating until it is not a duplicate
    while pick in numbers:
        pick = random.randint(1,60)

    # add to list
    numbers.append(pick)

# convert to ascending order
numbers.sort()

# display numbers
print("Your numbers are:", end=" ")

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

5. **Larger than n**

In a program, write a function called largerThanN that accepts two arguments: a list, and a number n. Assume that the list contains numbers. The function should return a list of all of the numbers in the list that are greater than the number n. Document your function using IPO notation.

Sample output:

```
print( largerThanN([3, 4, 5, 6, 7], 2) ) # [3, 4, 5, 6, 7]
print( largerThanN([-5, -6, -7, -8, -9], -3) ) # []
print( largerThanN([35.5, 99.33, 2.0, 8.78, -9], 10.5) )
# [35.5, 99.33]
```

```
# function: largerThanN
# input: list and a number (int)
# processing: display all values within list larger than n
# output: return a list with all values larger than n

def largerThanN(mylist, n):
    larger = []

    for v in mylist:
        if v > n:
            larger.append(v)

    return larger

print( largerThanN([3, 4, 5, 6, 7], 2) ) # [3, 4, 5, 6, 7]
print( largerThanN([-5, -6, -7, -8, -9], -3) ) # []
print( largerThanN([35.5, 99.33, 2.0, 8.78, -9], 10.5) ) # [35.5,
99.33]
```

ASCII Code Table

0	<NUL>	32	<SPC>	64	@	96	`	128	Ä	160	†	192	¿	224	‡
1	<SOH>	33	!	65	A	97	a	129	Å	161	°	193	¡	225	·
2	<STX>	34	"	66	B	98	b	130	Ç	162	¢	194	¬	226	‚
3	<ETX>	35	#	67	C	99	c	131	É	163	£	195	√	227	„
4	<EOT>	36	$	68	D	100	d	132	Ñ	164	§	196	ƒ	228	‰
5	<ENQ>	37	%	69	E	101	e	133	Ö	165	•	197	≈	229	Â
6	<ACK>	38	&	70	F	102	f	134	Ü	166	¶	198	Δ	230	Ê
7	<BEL>	39	'	71	G	103	g	135	á	167	ß	199	«	231	Á
8	<BS>	40	(	72	H	104	h	136	à	168	®	200	»	232	Ë
9	<TAB>	41	)	73	I	105	i	137	â	169	©	201	…	233	È
10	<LF>	42	*	74	J	106	j	138	ä	170	™	202		234	Í
11	<VT>	43	+	75	K	107	k	139	ã	171	´	203	À	235	Î
12	<FF>	44	,	76	L	108	l	140	å	172	¨	204	Ã	236	Ï
13	<CR>	45	-	77	M	109	m	141	ç	173	≠	205	Õ	237	Ì
14	<SO>	46	.	78	N	110	n	142	é	174	Æ	206	Œ	238	Ó
15	<SI>	47	/	79	O	111	o	143	è	175	Ø	207	œ	239	Ô
16	<DLE>	48	0	80	P	112	p	144	ê	176	∞	208	–	240	
17	<DC1>	49	1	81	Q	113	q	145	ë	177	±	209	—	241	Ò
18	<DC2>	50	2	82	R	114	r	146	í	178	≤	210	“	242	Ú
19	<DC3>	51	3	83	S	115	s	147	ì	179	≥	211	”	243	Û
20	<DC4>	52	4	84	T	116	t	148	î	180	¥	212	‘	244	Ù
21	<NAK>	53	5	85	U	117	u	149	ï	181	µ	213	’	245	ı
22	<SYN	54	6	86	V	118	v	150	ñ	182	∂	214	÷	246	ˆ
23	<ETB>	55	7	87	W	119	w	151	ó	183	∑	215	◊	247	˜
24	<CAN>	56	8	88	X	120	x	152	ò	184	∏	216	ÿ	248	¯
25	<EM>	57	9	89	Y	121	y	153	ô	185	π	217	Ÿ	249	˘
26	<SUB>	58	:	90	Z	122	z	154	ö	186	∫	218	⁄	250	˙
27	<ESC>	59	;	91	[	123	{	155	õ	187	ª	219	€	251	°
28	<FS>	60	<	92	\	124	|	156	ú	188	º	220	‹	252	¸
29	<GS>	61	=	93	]	125	}	157	ù	189	Ω	221	›	253	˝
30	<RS>	62	>	94	^	126	~	158	û	190	æ	222	fi	254	˛
31	<US>	63	?	95	_	127	<DEL>	159	ü	191	ø	223	fl	255	ˇ