CSE101 – Introduction to Computers
Python Programming Assignment # 2
(25 points, Submission due date: 5 April 2019)

Instructions

For each of the following problems, create an error free efficient Python program. Each program should be submitted in a separate Python file respectively that follows a particular naming convention. (E.g. The Python program for Question 1 should be in .py file with name Assign2Answer1.py. The Python program for question 2 should be in .py file with name Assign2Answer2.py. Include one or two input cases in your program. The program should execute properly in PyCharm).

Problems

Problem 1: (6 points)

Write a Python program that uses function ***def printAsteriskPattern(num_rows)*** to construct the following pattern, using a nested for loop. The function receives as input the number of rows (both, even and odd) in a pattern. In case the number of rows is odd, the number of asterisks in the middle row are: math.ceil(num_rows/2). In case the number of rows is even, there are two middle rows with the number of asterisks math.ceil(num_rows/2) as shown in the following:

```
>>>printAsteriskPattern(9)

*
* *
* * *
* * * *
* * * * *
* * * *
* * *
* *
*


>>>printAsteriskPattern(10)

*
* *
* * *
* * * *
* * * * *
* * * * *
* * * *
* * *
```

```
* *
*
```

Problem 2: (6 points)

A sieve function from PythonLabs takes as input any positive number and prints a list of prime numbers less than the given number. E.g.

```
>>> from PythonLabs.SieveLab import *
>>> sieve(50)
[2, 3, 5, 7, 11, 13, 17, 19, 23, 29, 31, 37, 41, 43, 47]
```

Write a function named ***def num_primes(num_list)*** that prints a table that shows how many primes are less than n for various values of n in a num_list. The argument to your function should be a list of numbers to use for n, and the output should be a set of lines that shows the value of n and the number of primes less than n. You can make use of PythonLabs sieve function in your program.

```
>>> num_primes ([10, 100, 1000, 10000, 100000])
10 4
100 25
1000 168
10000 1229
100000 9592
```

Problem 3: (6 points)

A prime pair is a set of two numbers that are both prime and that differ by 2. For example, 599 and 601 are a prime pair. Write a function named def ***prime_pairs(num)*** that prints all the prime pairs less than a specified value. You can make use of PythonLabs sieve function in your program.

```
>>> prime_pairs(50)
3 5
5 7
11 13
17 19
29 31
41 43
```

Problem 4: (7 points)

ASCII stands for American Standard Code for Information Interchange. Computers can only understand numbers, so an ASCII code is the numerical representation of a character such as 'a' or '@' or an action of some sort. Refer to the ASCII table shown in the following:

ASCII Table

Dec	Hex	Oct	Char	Dec	Hex	Oct	Char	Dec	Hex	Oct	Char	Dec	Hex	Oct	Char
0	0	0		32	20	40	[space]	64	40	100	@	96	60	140	`
1	1	1		33	21	41	!	65	41	101	A	97	61	141	a
2	2	2		34	22	42	"	66	42	102	B	98	62	142	b
3	3	3		35	23	43	#	67	43	103	C	99	63	143	c
4	4	4		36	24	44	$	68	44	104	D	100	64	144	d
5	5	5		37	25	45	%	69	45	105	E	101	65	145	e
6	6	6		38	26	46	&	70	46	106	F	102	66	146	f
7	7	7		39	27	47	'	71	47	107	G	103	67	147	g
8	8	10		40	28	50	(	72	48	110	H	104	68	150	h
9	9	11		41	29	51	)	73	49	111	I	105	69	151	i
10	A	12		42	2A	52	*	74	4A	112	J	106	6A	152	j
11	B	13		43	2B	53	+	75	4B	113	K	107	6B	153	k
12	C	14		44	2C	54	,	76	4C	114	L	108	6C	154	l
13	D	15		45	2D	55	-	77	4D	115	M	109	6D	155	m
14	E	16		46	2E	56	.	78	4E	116	N	110	6E	156	n
15	F	17		47	2F	57	/	79	4F	117	O	111	6F	157	o
16	10	20		48	30	60	0	80	50	120	P	112	70	160	p
17	11	21		49	31	61	1	81	51	121	Q	113	71	161	q
18	12	22		50	32	62	2	82	52	122	R	114	72	162	r
19	13	23		51	33	63	3	83	53	123	S	115	73	163	s
20	14	24		52	34	64	4	84	54	124	T	116	74	164	t
21	15	25		53	35	65	5	85	55	125	U	117	75	165	u
22	16	26		54	36	66	6	86	56	126	V	118	76	166	v
23	17	27		55	37	67	7	87	57	127	W	119	77	167	w
24	18	30		56	38	70	8	88	58	130	X	120	78	170	x
25	19	31		57	39	71	9	89	59	131	Y	121	79	171	y
26	1A	32		58	3A	72	:	90	5A	132	Z	122	7A	172	z
27	1B	33		59	3B	73	;	91	5B	133	[	123	7B	173	{
28	1C	34		60	3C	74	<	92	5C	134	\	124	7C	174	|
29	1D	35		61	3D	75	=	93	5D	135	]	125	7D	175	}
30	1E	36		62	3E	76	>	94	5E	136	^	126	7E	176	~
31	1F	37		63	3F	77	?	95	5F	137	_	127	7F	177	

In Python, you can use a function ***chr(num)*** to print character corresponding to a specific ASCII number. E.g.

>>> chr(65)

'A'

Similarly, a character can be converted to ASCII number using a function ***ord(character)***. E.g.

>>>ord('A')

65

Write a function named ***def asciiPyramid(num_lines)*** which will print a pyramid of ASCII characters for the specific number of lines. Check for the following condition: num_lines should be more than 0 and less than 14. The top of the pyramid is ascii character '!' which can be printed using function chr(33).

```
>>>asciiPyramid(6)
```

```
!
" #
$ % &
' ( ) *
+ , - . /
0 1 2 3 4 5

>>>asciiPyramid(13)
!
" #
$ % &
' ( ) *
+ , - . /
0 1 2 3 4 5
6 7 8 9 : ; <
= > ? @ A B C D
E F G H I J K L M
N O P Q R S T U V W
X Y Z [ \ ] ^ _ ` a b
c d e f g h i j k l m n
o p q r s t u v w x y z {
```