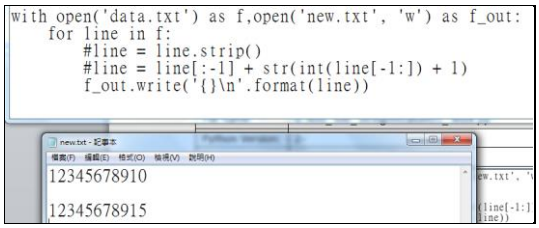
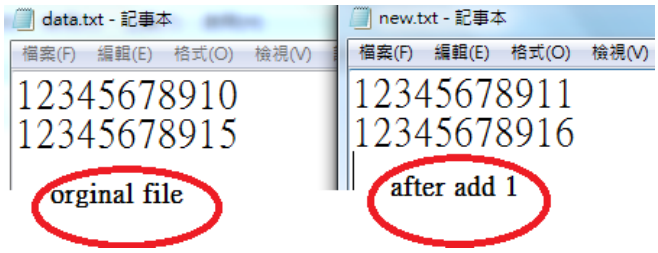
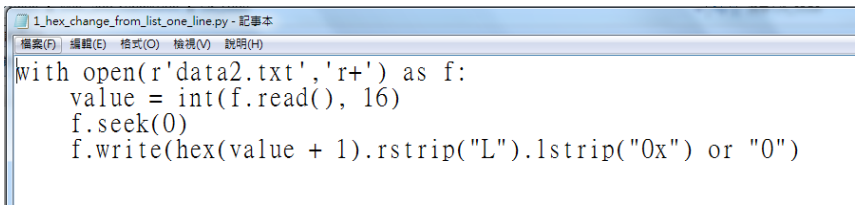
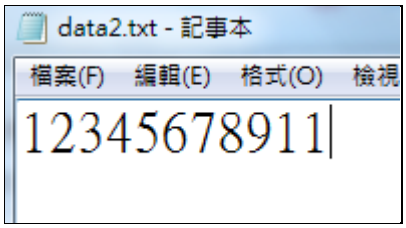
Description and note

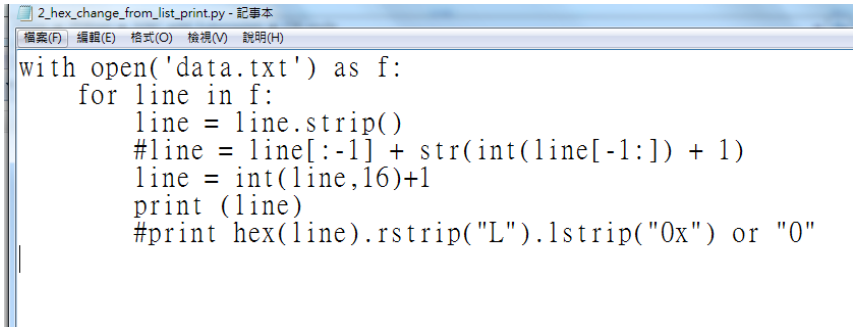
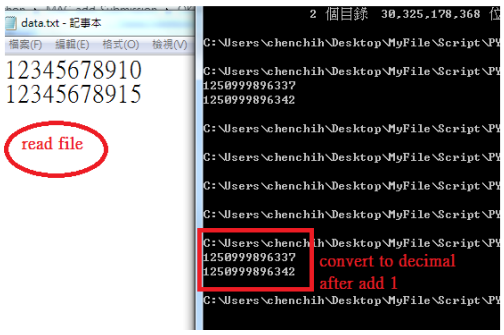
Description:	Read file then add 1 to it, export to a new file
File name:	0. Add_text_StringIntvalueto1_TestX.py
Python Version:	2
Note:	line.strip not put will skip each line 

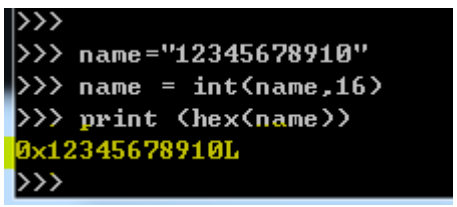
Code:	`with open('data.txt') as f,open('new.txt', 'w') as f_out:`
	`  for line in f:`
	`  line = line.strip()`
	`  line = line[:-1] + str(int(line[-1:]) + 1)`
	`  f_out.write('{}\n'.format(line))`

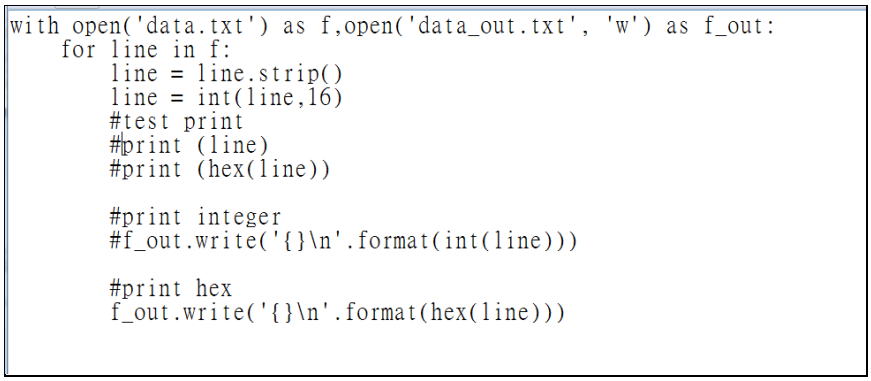
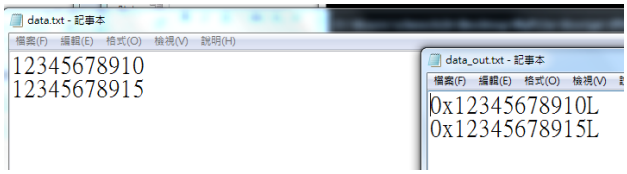
Output:	

Description:	Read file and add 1 to it and replace existed file
File name:	1_hex_change_from_list_one_line.py
Python Version:	2
Note:	
Code:	

Output:	

Description:	Read file into hex and 1 and convert and to decimal
File name:	2_hex_change_from_list_print.py
Python Version:	2
Note:	Example: Hex: (12345678910 +1)= 2468ACF1220 Decimal: 2501999792672
Code:	
Output:	

Description:	3 Read text file and print file_mult_line.py
File name:	file content is string, and convert to hex and print
Python Version:	2
Note:	

	Need to see 3-1 code to fix the –L issue
Code:	
Output:	

```
with open('data.txt') as f,open('data_out.txt', 'w') as f_out:
    for line in f:
        line = line.strip()
        line = int(line,16)
        #test print
        #print (line)
        #print (hex(line))

        #print integer
        #f_out.write('{}\n'.format(int(line)))

        #print hex
        f_out.write('{}\n'.format(hex(line)))
```

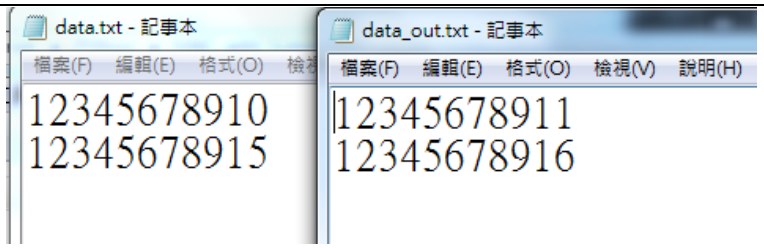

Description:	This is fix previous issue and add 1 to it If value more than 10 will print L in front of it.
File name:	3-1 Read hex text file and print file_1_mult_line.py
Python Version:	2
Note:	Need to add this, else it will print L f_out.write('{}\n'.format(hex(line).rstrip("L").lstrip("0x") or "0"))
Code:	

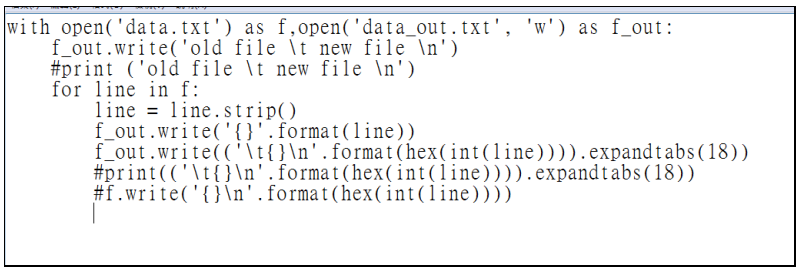
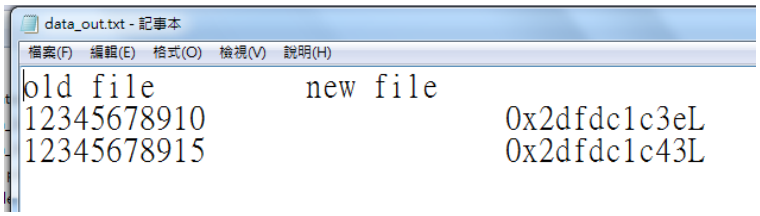
```
with open('data.txt') as f,open('data_out.txt', 'w') as f_out:
    for line in f:
        line = line.strip()
        line = int(line,16)+1
        #test print
        #print (line)
        #print (hex(line))

        #print integer
        #f_out.write('{}\n'.format(int(line)))

        #print hex
        #f_out.write('{}\n'.format(hex(line)))

        #If hex used 10 char will occur L in end, remove L(end) and 0x(begin)
        f_out.write('{}\n'.format(hex(line).rstrip("L").lstrip("0x") or "0"))
```

Output:	

Description:	Print two row, one old value, and one new value
File name:	4. read_file_and write_new_file.py
Python Version:	
Note:	f_out.write(('\t{}\n'.format(hex(int(line)))).expandtabs(18))
Code:	
Output:	

```
with open('data.txt') as f,open('data_out.txt', 'w') as f_out:
    f_out.write('old file \t new file \n')
    #print ('old file \t new file \n')
    for line in f:
        line = line.strip()
        f_out.write('{}'.format(line))
        f_out.write(('\t{}\n'.format(hex(int(line)))).expandtabs(18))
        #print(('\t{}\n'.format(hex(int(line)))).expandtabs(18))
        #f.write('{}\n'.format(hex(int(line))))
```