EECS 1210 Introduction to Programming in Python
Spring Semester 2024
Prof. Pai H. Chou

Self-Check 4

Answer the following questions to check your understanding of your material. Expect the same kind of questions to show up on your tests.

1. Definitions and Short Answers

1. Given the command shown on the lecture slide
   `$ uniq mary.txt`
   What is
   - the **prompt**? $
   - the **program name**? uniq
   - the **command-line argument**? mary.txt
2. What does the `uniq` program do? 將重複文字行過濾
3. What does the `cat` program do?把檔案串連接後傳到基本輸出
4. What does the `grep` program do?搜尋某檔案，找出”要搜尋的字串”的那行顯示出來
5. Is it possible that `uniq` and `cat` produce the same output? How?
   檔案不要有重覆行，只串接一個檔案
6. Given the command shown on the lecture slide
   `$ grep class myfile.py`
   What is the purpose of
   - `class` 要找的字串是 class
   - `myfile.py` 要找 myfile.py 裡具 class 的行
7. Given the command
   `$ cat *.py`
   What is the meaning of `*.py`? 所有以.py結尾的檔案
8. What does the following command do?
   `$ python3 prog.py` 用 Python3 執行 prog.py
9. What is a **shbang** in a Python program? Where is it placed inside a Python program?
   目的是指出該用什麼程式執行該 Script
   需放在 Script 的最前面
10. What does the command do:
    `$ chmod +x prog.py`
    賦予 prog.py 執行權限
11. What is the value of
    - `len([3, 7, 2, 0, 8])` 5
    - `len(['hello', 'world', 'goodbye'])` 3
    - `len('admin')` 5

12. Suppose you run the command

```
$ python3 showargs.py  hello  world  goodbye
```

Inside the `showargs.py` program, suppose you have

```
import sys
```

- ○ What is the value of `sys.argv`? ['showargs.py', 'hello', 'world', 'goodbye']
- ○ What is the value of `len(sys.argv)`? 4
- ○ What is the value of `sys.argv[1:]`? ['hello', 'world', 'goodbye']

13. If the command `$ python3 showargs.py hello world` is used to run the Python program, what is printed by the statement

```
import sys
sys.stderr.write('cannot open input file %s\n' % sys.argv[1])
```

? cannot open input file hello

14. If the file `mary.txt` contains the following lines

```
Mary had a little lamb
little lamb, little lamb
Mary had a little lamb
its fleece was white as snow
```

what is the **value** of `L` after executing the following statements?

```
fh = open('mary.txt', 'r')
L = fh.readlines()
fh.close()
```

?

['Mary had a little lamb\n', 'little lamb, little lamb\n', 'Mary had a little lamb\n', 'its fleece was white as snow\n']

15. What is the purpose of `end=''` in the statement

```
print(line, end='')
```

?

在印完 line 的內容後以空字元結尾，而非預設的換行符號'\n'

16. Explain why

```
'hello'.find('e')
```

results in the integer value of 1, while

```
'hello'.find('a')
```

results in `-1`.

因為 hello 裡面有 e 沒有 a

17. Rewrite the **string literal** `"hello, I'm John."` using

- ○ single quotes 'hello, I\'m John'
- ○ triple single quotes '''hello, I'm John'''
- ○ triple double quotes
  instead of double quotes. """hello, I'm John"""

18. Rewrite the string literal `'she says, "This is great!" and left'` using

- ○ double quotes "she says, \"This is great!\" and left"
- ○ triple single quotes '''she says, "This is great!" and left'''
- ○ triple double quotes
  instead of single quotes. """she says, "This is great!" and left"""

19. Rewrite the string literal `'\\n means newline'` using a **raw string**.
    r'\n means newline'
20. After executing the statement
    `t = 'hello' "world"`
    What is the value of `t`?
    helloworld
21. What is the value of
    - `len("hello")` 5
    - `len("I\tam\there")` 9
      ```
      >>> print("I\tam\there")
      I       am      here
      ```
    - `len('McDonald\'s')` 10
22. Rewrite the following triple-quoted string literal using a non-triple-quoted string literal
    ```
    sourceCode = '''<html>
    <body>Welcome</body>
    <html>'''
    ```
    - on one single line
      '<html>\n<body>Welcome</body>\n<html>'
    - on three separate lines
      '<html>\n\
      <body>Welcome</body>\n\
      <html>'
23. Assume
    ```
    month = 7
    day = 4
    year = 2019
    ```
    How do you format the date using % formatting so that it appears as strings (expressed as string literals)
    - `'7/4/2019'`
      '{:d}/{:d}/{:d}'
    - `'07/04/2019'`
      '{:02d}/{:02d}/{:d}'
24. What is the value of
    - `'%9.2f' % 13.5`
      `'    13.50'`
    - `'%9.2f' % 123456789.0193`
      `'123456789.02'`
25. What is the meaning of `5e2`? What is its data type?
    500 float
26. What is the value of `5e-2`?
    0.05
27. What is the value of `'%c' % 100`, given that `ord('a')` has the value of 97? d
28. What is the format string S such that `S.format(month, day, year)` is equivalent to the traditional formatting of

`'%d/%d/%d' % (month, day, year)`
?
f'{month}/{day}/{year}'

29. What is the value of the expression
    `'one {0}, two {0}s, three {0}s'.format('apple')` ?
    'one apple, two apples, three apples'
30. What is the format string S such that
    `S.format(12)`
    evaluates to the string
    `'12 decimal is 0c hex and 14 octal'` `# 需先定義 n`
    `S = '%d decimal is %02x hex and %o octal' % (n,n,n)`
31. What is the value of the expression
    `'lastname {1}, firstname {0}'.format('John', 'Smith')`
    ? 'lastname Smith, firstname John'
32. Rewrite the following expressions as **f-string**:
    - `'%d/%d/%d' % (month, day, year)`
      `f"{'%d'%month}/{'%d'%day}/{'%d'%year}"`
    - `'{:02d}/{:02d}/{:04d}'.format(month, day, year)`
      `f"{'%02d'%month}/{'%02d'%day}/{'%04d'%year}"`
33. What is the value of the expression
    - `'www.nthu.edu.tw'.split('.')`
      `['www', 'nthu', 'edu', 'tw']`
    - `'Mary had a\nlittle lamb'.split()`
      `['Mary', 'had', 'a', 'little', 'lamb']`
34. Suppose you type the unix command wc (lightblue) and get the output (lightgreen) as shown below:

    ```
    $ wc mult.py
           9      32     249  mult.py
    ```

    What are the meanings of 9, 32, and 249?
    9 lines, 32 words, 249 characters
35. What is the value of the expression
    - `'(' + ')('.join(['a', 'b', 'c', 'd']) + ')'`
      `'(a)(b)(c)(d)'`
    - `''.join('Mary had a little lamb'.split())`
      'Maryhadalittlelamb'
36. Assume you have
    `import string`
    What is the value of
    - `string.punctuation`
      ``'!"#$%&\'()*+,-./:;<=>?@[\\]^_`{|}~'``
    - `string.digits`
      `'0123456789'`
    - `string.ascii_lowercase`

    `'abcdefghijklmnopqrstuvwxyz'`

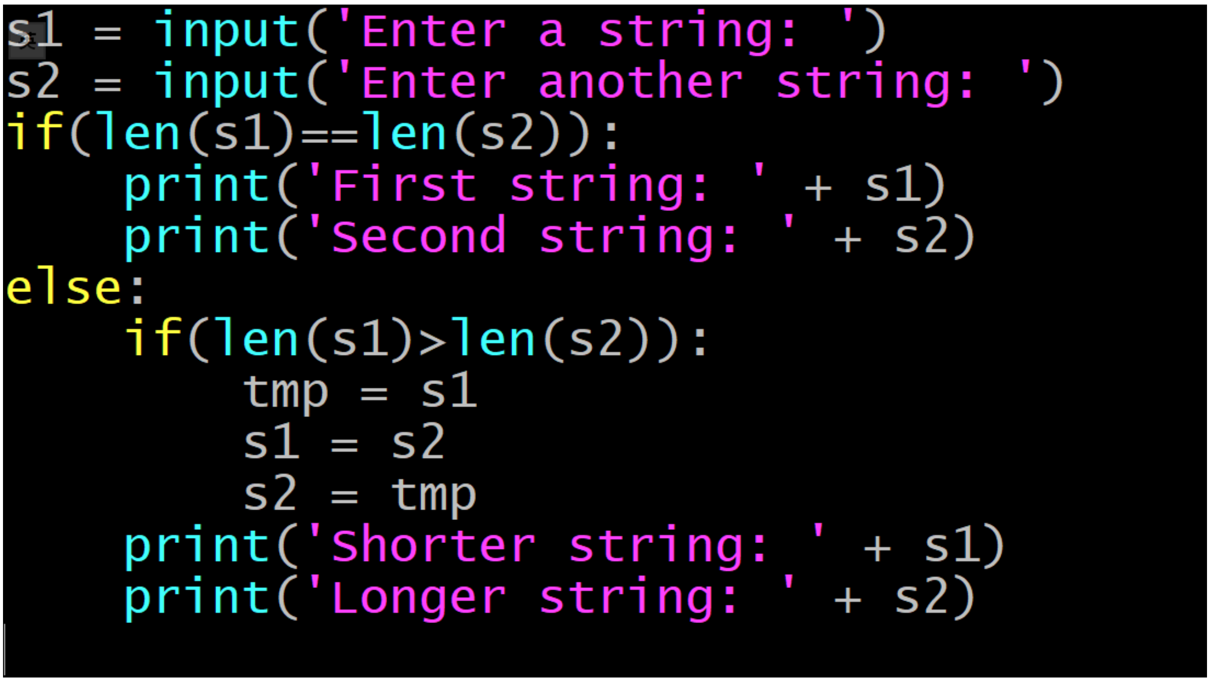
    ○ `string.whitespace`
      `' \t\n\r\x0b\x0c'`
    ○ `string.printable`
      ``'0123456789abcdefghijklmnopqrstuvwxyzABCDEFGHIJKLMNOPQRSTUVWXYZ!"#$%&\'()*+,-./:;<=>?@[\\]^_`{|}~ \t\n\r\x0b\x0c'``

2. Programming Exercises

1. Write a program that prompts the user to input two strings and reports the two strings' lengths, by reporting the shorter string first. But if they are of the same length then keep them in the original order. For example, (blue text = typed input, green highlight = program printout)

   ```
   $ python3 compstr.py
   Enter a string: Great
   Enter another string: job
   Shorter string: job (length 3)
   Longer string: Great (length 5)
   $ python3 compstr.py
   Enter a string: Mary
   Enter another string: lamb
   First string: Mary (length 4)
   Second string: lamb (length 4)
   $
   ```

   Note that in case the strings are of different lengths, the program says `Shorter` and `Longer`, but in case the strings are of equal length, the program says `First` and `Second`. You don't actually print in color… the letters are colored for illustration purpose only.



```
s1 = input('Enter a string: ')
s2 = input('Enter another string: ')
if(len(s1)==len(s2)):
    print('First string: ' + s1)
    print('Second string: ' + s2)
else:
    if(len(s1)>len(s2)):
        tmp = s1
        s1 = s2
        s2 = tmp
    print('Shorter string: ' + s1)
    print('Longer string: ' + s2)
```

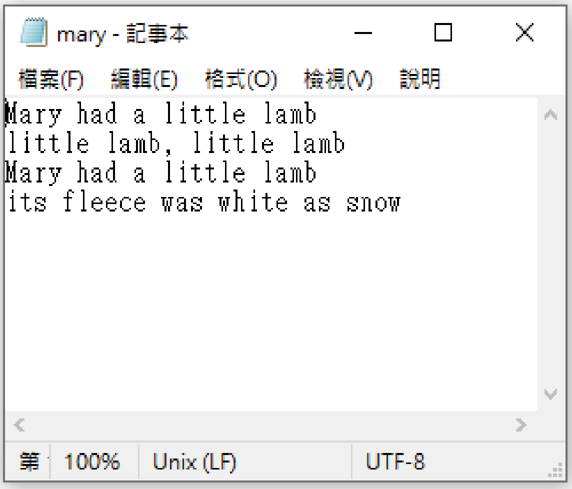
2. Write a Python program named catn.py by modifying the template code to implement the unix utility command cat with -n option, which adds the line number in front of every line of a file.
   a. First version: support the command with **optional** -n flag and **one file**. Note that the line number is formatted

```
$ python3 catn.py mary.txt
Mary had a little lamb
little lamb, little lamb
Mary had a little lamb
its fleece was white as snow
$ python3 catn.py -n mary.txt
     1  Mary had a little lamb
     2  little lamb, little lamb
     3  Mary had a little lamb
     4  its fleece was wvhite as snow
$
```

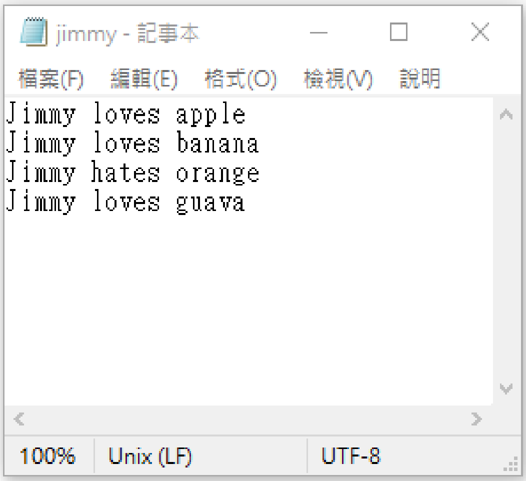
   版本二可實現版本一之功能，故版本一不實作

   b. Second version: handles **one or more files** with optional -n flag. In case of multiple files, the line number restarts from 1.

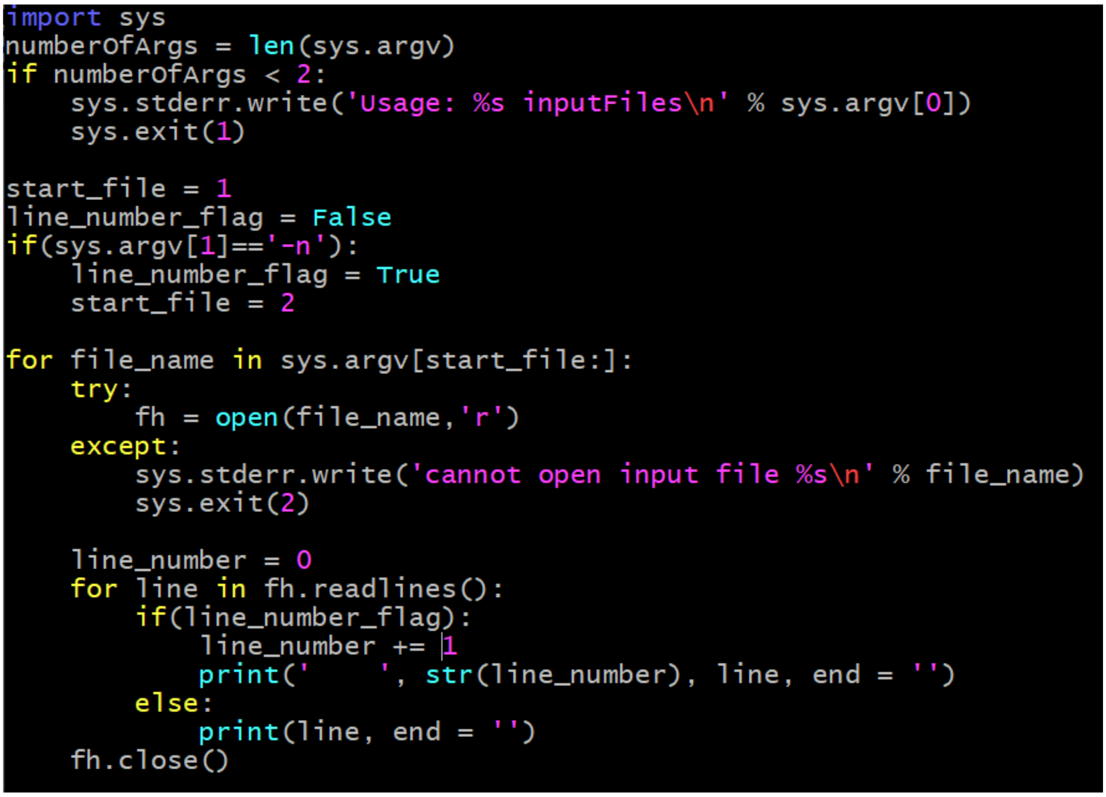
版本二之實作如下圖



```
Mary had a little lamb
little lamb, little lamb
Mary had a little lamb
its fleece was white as snow
```



```
Jimmy loves apple
Jimmy loves banana
Jimmy hates orange
Jimmy loves guava
```

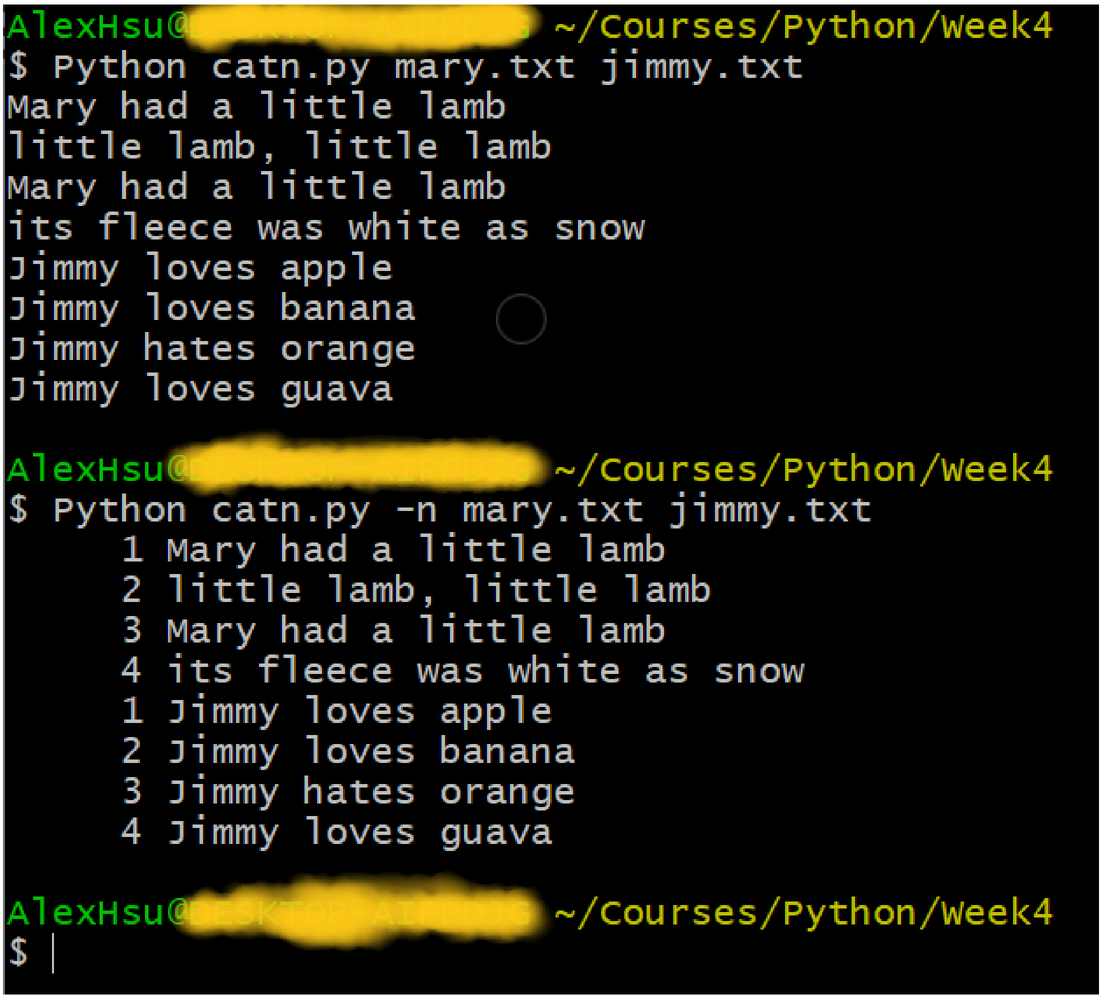


```
import sys
numberOfArgs = len(sys.argv)
if numberOfArgs < 2:
    sys.stderr.write('Usage: %s inputFiles\n' % sys.argv[0])
    sys.exit(1)

start_file = 1
line_number_flag = False
if(sys.argv[1]=='-n'):
    line_number_flag = True
    start_file = 2

for file_name in sys.argv[start_file:]:
    try:
        fh = open(file_name,'r')
    except:
        sys.stderr.write('cannot open input file %s\n' % file_name)
        sys.exit(2)

    line_number = 0
    for line in fh.readlines():
        if(line_number_flag):
            line_number += 1
            print('    ', str(line_number), line, end = '')
        else:
            print(line, end = '')
    fh.close()
```



```
AlexHsu@[redacted] ~/Courses/Python/Week4
$ Python catn.py mary.txt jimmy.txt
Mary had a little lamb
little lamb, little lamb
Mary had a little lamb
its fleece was white as snow
Jimmy loves apple
Jimmy loves banana
Jimmy hates orange
Jimmy loves guava

AlexHsu@[redacted] ~/Courses/Python/Week4
$ Python catn.py -n mary.txt jimmy.txt
     1 Mary had a little lamb
     2 little lamb, little lamb
     3 Mary had a little lamb
     4 its fleece was white as snow
     1 Jimmy loves apple
     2 Jimmy loves banana
     3 Jimmy hates orange
     4 Jimmy loves guava

AlexHsu@[redacted] ~/Courses/Python/Week4
$ 
```