SC4

1. Definitions and Short Answers

1. Given the command shown on the lecture slide

   $ uniq mary.txt

   What is

   - the **prompt**?

   > 💡 $

   - the **program name**?

   > 💡 The program name is `uniq`, which is a command-line utility that filters adjacent matching lines from input.

   - the **command-line argument**?

   > 💡 The command-line argument is `mary.txt`, which is the input file that `uniq` will process to remove any adjacent duplicate lines.

2. What does the uniq program do?
3. What does the cat program do?

   > 💡 The `cat` command is a commonly used utility in Unix-like operating systems that is used to concatenate and display the contents of files. The name "cat" stands for "concatenate".

4. What does the grep program do?

> 💡 The `grep` command is a powerful Unix utility that is used to search for patterns in text files or streams. The name "grep" stands for "global regular expression print".

5. Is it possible that uniq and cat produce the same output? How?

> 💡 If a file has no adjacent duplicate lines, then running `uniq` on the file will produce the same output as running `cat` on the file.

6. Given the command shown on the lecture slide

   $ grep class myfile.py

   What is the purpose of

   - class

   > 💡 pattern

   - myfile.py

   > 💡 input file

7. Given the command

   $ cat *.py

   What is the meaning of *.py?

   > 💡 wildcard

8. What does the following command do?

   $ python3 prog.py

> 💡 The command `$ python3 prog.py` is used to execute a Python program named `prog.py` using the Python 3 interpreter.

9. What is a **shbang** in a Python program? Where is it placed inside a Python program?

> 💡 A "shebang" (also called "hashbang" or "sha-bang") is a special line that appears at the beginning of a Unix/Linux shell script or a Python script that tells the operating system which interpreter to use to execute the script.

> 💡 In Python, the shebang line (also called the "shebang directive") is a line that starts with `#!` (hash and exclamation mark characters) followed by the path to the Python interpreter.

10. What does the command do:

    $ chmod +x prog.py

> 💡 讓prog.py變成可執行檔

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()?

    💡 The variable `L` will contain a list of strings, where each string corresponds to a line in the file `mary.txt`. Specifically, `L` will be:

    ['Mary had a little lamb\n',
    'little lamb \n',
    '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='')?

    💡 In Python, the `print()` function is used to display output on the screen or write it to a file. By default, the `print()` function adds a newline character (`\n`) to the end of the output it displays, which means that each call to `print()` starts a new line of output.

> 💡 In the statement `print(line, end='')`, the `end=''` argument is used to modify the default behavior of `print()` so that it does not add a newline character at the end of the output it displays. Instead, it ends the output with an empty string (`''`).

16. Explain why

    'hello'.find('e') results in the integer value of 1, while

    'hello'.find('a') results in -1.

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

    > 💡 """hello, I'm John."""

instead of double quotes.

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

> 💡 """'she says, "This is great!" and left'"""

instead of single quotes.

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?

> 💡 The value of `t` is a string `'helloworld'`.

> 💡 In Python, adjacent string literals are automatically concatenated into a single string. This is known as string concatenation or string literal concatenation. When two string literals appear one after another without any intervening operator or punctuation, they are automatically concatenated into a single string.

21. What is the value of

    - len("hello")

💡 5

- len("I\tam\there")

💡 9

- 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

```
sourceCode = "<html>\n<body>Welcome</body>\n<html>"'
```

- on three separate lines

```
sourceCode = "<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'

> 💡 print(”%d/%d/%d” % (month, day, year))
> or
> print('%d/%d/%d' % (month, day, year))

- '07/04/2019'

> 💡 print('%02d/%02d/%d' % (month, day, year))

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?

> 💡 5*10^2
> The data type of `5e2` is a floating-point number, which is represented in Python using the `float` type.

26. What is the value of 5e-2?

💡 5*10^-2

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) ?

```
a = '{}/{}/{}'.format(2, 27, 2023)
print(a)
print('%d/%d/%d' % (2, 27, 2023))
```

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'

💡 '{0} decimal is {0:x} hex and {0:o} octal'.format(12)

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"{month}/{day}/{year}”

    - '{:02d}/{:02d}/{:04d}'.format(month, day, year)

    💡 f"{month:02d}/{day:02d}/{year:04d}”

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?

> 💡 In the output of the `wc` command, there are three numbers displayed for a given file. These numbers represent the count of lines, words, and bytes (characters) in the file, respectively.
>
> So in the output `$ wc mult.py`,
>
> - `9` represents the number of lines in the file `mult.py`
> - `32` represents the number of words in the file `mult.py`
> - `249` represents the number of bytes (characters) in the file `mult.py`

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
    - string.ascii_lowercase
    - string.whitespace
    - string.printable