Chapter 8 More on Strings and Special Methods

1.

a. s1 == s2	True	l. 4 * s4	totototo
b. s2.count('o')	3	m. len(s1)	17
c. id(s1) == id(s2)	True	n. max(s1)	y
d. id(s1) == id(s3)	False	o. min(s1)	a space char
e. s1 <= s4	True	p. s1[-4]	t
f. s2 >= s4	False	q. s1.lower()	welcome to python
g. s1 != s4	True	r. s1.rfind('o')	15
h. s1.upper()	WELCOME TO	s. s1.startswith("Welco")	True
i. s1.find(s4)	8	t. s1.endswith("on")	True
j. s1[4]	o	u. s1.isalpha()	False
k. s1[4 : 8]	ome	v. s1 + s2	

2

All correct except

```
s3 = s1 - s2   # Incorrect
```

3

```
Welcome to Python
Welcabcme tabc Pythabcn
```

4

```
(a) isEqual = (s1 == s2)

(b) isEqual = (s1.upper() == s2.upper())

(c) b = s1.startswith("AAA")

(d) b = s1.endswith("AAA")

(e) x = len(s1)

(f) x = s1[0]

(g) s3 = s1 + s2

(h) s3 = s1[1:]

(i) s3 = s1[1: 5]

(j) s3 = s1.lower()
```

(k) s3 = s1.upper()

(l) s3 = s1.strip()

(m) s3 = s1.replace('e', 'E')

(n) x = s1.find('e')

(o) x = s1.rfind('abc')

5

No. Strings are immutable.

6
0

7
s.islower() or s.isupper()

8
s.isalpha()

9. You can define methods for operators. This is called operator overloading.

10. The methods for operators +, -, *, /, %, ==, !=, <, <=, >, >= are __add__(self, other), __sub__(self, other), __mul__(self, other), __div__(self, other), __mod__(self, other), __eq__(self, other), __ne__(self, other), __lt__(self, other), __le__(self, other), __gt__(self, other), and __ge__(self, other).

11. Yes
12. Yes