

Exercise walk-through

Implementing String Algorithms ■ 2.10

eXercise

You must be able to

- **traverse** 遍历 a String by looping over its indices from 0 to `length - 1`
- use `charAt` and `substring` inside a loop
- count characters or find a pattern one position at a time

Method — Traversing a String

Loop over the valid indices 0 to `length() - 1` and read each character with `charAt`:

```
kfor+w p(k+ktint+w ni+w o+=w l+m+mi0p;+w ni+w o+w nsp.n+nalengthp(p)p;+w n
+w      k+ktchar+w nc+w o+=w nsp.n+nacharAtp(nip)p;+w      c+c1// the character
p
```

Method — Counting and searching

By examining one position at a time, you can **count** particular characters or **find** a pattern within the String.

Practice 2.1 ■ 1 mark

State the range of indices used to traverse a String `s`.

Solution 2.1

Method: Traversing a String

Worked steps

from 0 to `s.length() - 1` **B1**.

Practice 2.2 ■ 1 mark

State what `s.charAt(0)` returns.

Solution 2.2

Worked steps

the first character of the String **B1**.

Practice 2.3 ■ 1 mark

For `s = "apple"`, state the value of `s.charAt(1)`.

Solution 2.3

Worked steps

'p' **B1**.

Exam-style 3.1 ■ 1 mark

To visit every character of a String, loop from 0 to

A length

B length - 1

C length + 1

D 1

Solution 3.1

Method: Traversing a String

A length

B length - 1



C length + 1

D 1

Worked steps

B.

Exam-style 3.2 ■ 1 mark

`s.charAt(i)` returns

- A** a substring
- B** the character at index `i`
- C** the length
- D** the index `i`

Solution 3.2

- A a substring
- B the character at index i
- C the length
- D the index i



Worked steps

B.

Exam-style 3.3 ■ 1 mark

```
String s = "hello";
```

- (a) State `s.length()`.
- (b) State `s.charAt(4)`.
- (c) State the last valid index.

Solution 3.3

Method: Traversing a String

Worked steps

(a) 5 **B1**. (b) 'o' **B1**. (c) 4 **B1**.