Solutions

Marking: **M1** a correct method, **A1** the correct answer, **B1** an independent mark.

Each answer shows the steps, then the **result**.

2.1

`IDENT(count)`, `ASSIGN`

`IDENT(a1)`, `PLUS`, `IDENT(b2)`, `SEMI`

6 tokens; spaces make none

2.2

`ifx`: one `IDENT`, because the longer match wins over the keyword `if`

`if`: keyword `IF`, because the keyword rule comes first

2.3

```
0|[1-9][0-9]*
```

```
[A-Za-z_][A-Za-z0-9_]*
```

```
[0-9]+\.[0-9]+
```

2.4

longest match gives `NUM(12)` then `IDENT(abc)`

many languages report an error instead, because a number must not run straight into a letter

3.1

(a) start: `<`→s1, `=`→s4 (accept `EQ`); s1 accepts `LT`; s1: `=`→s2 (accept `LE`), `>`→s3 (accept `NE`); every other character from s1 ends `LT`

(b) `<`→s1, `=`→s2, then `5` has no move: emit `LE`; after `<` the lexer cannot tell `<` from `<=` until it sees the next character

(c) one pass over the input, in linear time; the DFA remembers the last accepting state, which gives longest match for all classes at once

3.2

(a) `IDENT(x)`, `ASSIGN`; `STRING("a // b")`, `SEMI`, and the comment makes no token

(b) it cuts the string in half: `"a` is left without its closing quote

(c) the depth can grow without limit, so it is not regular; the lexer keeps a depth counter

3.3

(a) `NUM(3.5)`, `STAR`, `LPAREN`, `IDENT(x1)`; `PLUS`, `NUM(42)`, `RPAREN`: 7 tokens

(b) 2×10^6 moves; $2 \times 10^6 / 10^8 = \mathbf{0.02}$ s

(c) give the line and column of the bad character, and say what was expected: “a digit after the decimal point”

exercises.lol exercises.com.cn

(d) for a small token set, a hand-written lexer is short and gives friendly errors; a generator suits a large or changing token set; for a calculator, write it by hand