Writing pseudocode

Ref As Pseudocode

Keywords, identifiers and case

Pseudocode is a way to write an algorithm that looks like a program but is not tied to one programming language. In the 9618 exam, every algorithm in a question is printed in pseudocode, and many answers must be written in it. Cambridge publishes one guide that sets the rules, and this reference follows it.

You do not need perfect pseudocode to get the marks. You do need pseudocode that a marker can read **without guessing**. The rules below make that easy.

Keywords are in capitals

A **keyword** is a word with a fixed meaning, such as `IF`, `WHILE`, `DECLARE` or `OUTPUT`. Keywords are always written in **upper case** upper case .

```
DECLARE Total : INTEGER
Total ← 10 + 5
OUTPUT Total
```

Here `DECLARE`, `INTEGER` and `OUTPUT` are keywords. `Total` is a name that we chose.

Identifiers

An **identifier** is the name you give to a variable, a constant, a procedure or a function. The 9618 rules are:

- It **starts with a letter** (A–Z or a–z), never a digit.
- After the first letter it may contain letters, digits (0–9) and the underscore `_`.
- No spaces, no accented letters, and no other symbols.
- It must **not** be a keyword. `Output` or `Case` cannot be a variable name.

Identifier	Allowed?	Why
`TotalScore`	yes	letters only
`Player2`	yes	a digit after the first letter is fine
`max_value`	yes	9618 allows the underscore and a small first letter
`2ndPlace`	**no**	starts with a digit
`Total Score`	**no**	contains a space
`WHILE`	**no**	it is a keyword

Cambridge writes identifiers in **mixed case** mixed case : each new word starts with a capital, as in `NumberOfPlayers`. This is the style you will see in every paper.

exercises.lol exercises.com.cn

■ Choose names that explain themselves

Good names make your answer easier to mark. `Average` says what the value means; `A` does not. Single letters are fine only where they are a normal habit: `i` and `j` for array indices, `X` and `Y` for coordinates.

■ Identifiers are case-insensitive

`Count`, `COUNT` and `count` are treated as the **same** name. So never use two names that differ only in case, such as `CountDown` and `Countdown`. A marker will see one variable, and your algorithm will not mean what you wanted.

```
DECLARE Count : INTEGER
Count ← 3
count ← count + 1
OUTPUT COUNT
```

This prints `4`: all three spellings reach one variable. It runs, but it is bad style. Pick one spelling and keep it.

Common mistakes

- Writing keywords in small letters, such as `if` or `output`. Write `IF` and `OUTPUT`.
- Using a keyword as a name, for example `DECLARE String : STRING`.
- Putting a space inside a name: write `FirstName`, not `First Name`.

Indentation and comments

■ Indentation shows what is inside what

Indentation means moving a line to the right. In pseudocode it shows that a line belongs **inside** the statement above it, for example inside a loop or an `IF`. The 9618 guide usually indents by **three spaces**.

```
DECLARE Counter : INTEGER
FOR Counter ← 1 TO 3
   OUTPUT "Line ", Counter
   IF Counter = 3 THEN
      OUTPUT "Done"
   ENDIF
NEXT Counter
```

- `OUTPUT "Line "` and the `IF` are inside the `FOR` loop, so they move in 3 spaces.
- `OUTPUT "Done"` is inside the `IF`, so it moves in 3 more.
- `ENDIF` lines up with its `IF`; `NEXT` lines up with its `FOR`.

The program would run the same with no indentation at all. But a marker reads the layout to see where each block starts and ends, so **always indent**. An answer with

no indentation is hard to follow and may lose marks when the structure is unclear.

Comments

A **comment** is a note for the human reader. The computer ignores it. A comment starts with two forward slashes `//` and runs to the end of the line.

```
// Work out the area of a rectangle
DECLARE Width : INTEGER
DECLARE Height : INTEGER
Width ← 4
Height ← 7
OUTPUT Width * Height   // area in square metres
```

- Usually a comment goes on its **own line**, before the code it explains, at the same indentation.
- A short comment may go at the end of one line, as in the last line above.
- A comment over several lines needs `//` at the start of **each** line.

Line numbers

Some questions print **line numbers** line numbers on the left, so that a question can say "line 07 has an error". The numbers are **not** part of the pseudocode. When you correct a line, write the line number and the new code, for example:

```
Line 07:  WHILE Index <= 10
```

Practice

Find what is wrong with each line. Then open the answer.

1. `DECLARE 1stName : STRING`

2. `declare Total : INTEGER`

3. `DECLARE Average Mark : REAL`

<details> <summary>Model answer</summary>

1. The identifier starts with a digit. Use `FirstName`.
2. The keyword must be in capitals: `DECLARE`.
3. An identifier cannot contain a space. Use `AverageMark`.

</details>

Common mistakes

- Forgetting to indent the body of a loop or an `IF`.
- Ending a block at the wrong level: `ENDIF` must line up with its own `IF`.
- Writing a comment with one slash `/`. One slash means **divide**; a comment needs `//`.