



F5.3 Pseudocode and the exam reference language

Algorithms · GCSE · OCR J277 2.1.2, AQA 8525 3.1.1, Edexcel 1CP2 1.2.2 ·
about 15 min

What this lesson is about

Reading and writing your board's pseudocode, and translating it into Python.

- The same algorithm, three ways
- Translating line by line
- Writing your own pseudocode
- Python as pseudocode

Questions

5 marks in all

1. What is pseudocode? [1 mark]

- ☐ A A way of writing an algorithm that looks like code but is not tied to one language
- ☐ B A programming language that runs on robots
- ☐ C Code with errors in it
- ☐ D Comments in a program

Answer: A. Pseudocode describes an algorithm clearly without the strict rules of a real language.

2. What is `17 MOD 5` in Python? Write the Python expression. [1 mark]

Answer: `17 % 5`. MOD in pseudocode is % in Python.

3. `for i = 1 to 5` in OCR's Exam Reference Language. How many times does the loop run? [1 mark]

Answer: 5. The Reference Language includes the end number, unlike Python's range.

4. Which Python line is the same as AQA's `x ← x + 1`? [1 mark]

- ☐ A `x = x + 1`
- ☐ B `x == x + 1`
- ☐ C `x <- x + 1`
- ☐ D `x + 1 = x`

Answer: A. ← is assignment in AQA pseudo-code; Python uses =.

5. Which Python loop is the same as `for i = 1 to 4`? [1 mark]

- ☐ A `for i in range(1, 5):`
- ☐ B `for i in range(1, 4):`
- ☐ C `for i in range(4):`
- ☐ D `for i in range(0, 4):`

Answer: A. range stops before its end number, so the end must be 5 to include 4.

The task: steps from pseudocode

Translate the algorithm at the top of this lesson into Python, from the pseudocode for your board. The robot should step towards the wall in 10 cm steps while it is more than 30 cm away, then print whether it took an even or odd number of steps, in exactly the form `even number of steps: 4`.

```
# the two lines every program starts with: the commands, then the robot
from bugbot import *
connect()

steps = 0
```

The hint students can ask for: Follow the pseudocode line by line: count the steps in a loop that runs while the wall is far enough away. At the end, test whether the count divides by two exactly to choose the message.

A solution

```
from bugbot import *
connect()
steps = 0
while distance() > 30:
    forward(50, distance=10)
    steps = steps + 1
if steps % 2 == 0:
    print("even number of steps: " + str(steps))
else:
    print("odd number of steps: " + str(steps))
```

Any program that meets the task's checks is marked correct in the simulator; this is one way, not the only way.