ANSWERS

F4.6 Project: the patrol

Functions and structured code · GCSE · about 25 min

What this lesson is about

A complete program built a function at a time: patrol to the wall and come home.

- The brief
- Decompose it
- Piece 1: one step
- Piece 2: the patrol
- Piece 3: coming home
- Check the whole program
- Where next

Questions

5 marks in all

1. Why test `step(n)` on its own before writing the loop? [1 mark]

- ☐ **A** Mistakes are easiest to find while the program is small
- ☐ **B** Python needs functions to be run once first
- ☐ **C** It charges the battery
- ☐ **D** The loop will not work otherwise

> **Answer:** A. Building a program a piece at a time means each piece is checked while it is still short.

2. What does a name written in capitals, like `STOP_AT = 30`, tell someone reading the program? [1 mark]

- ☐ **A** It is set once and not changed while the program runs
- ☐ **B** Python will not let it change
- ☐ **C** It is a function
- ☐ **D** It is printed automatically

> **Answer:** A. Capitals are a convention for values set once. Python does not enforce it; it is a message to people.

3. What does this program print? [1 mark]

```
n = 1
while n <= 3:
    print(f"step {n}")
    n = n + 1
print("stopped after", n - 1, "steps")
```

> **Answer:**
>
> ```
> step 1
> step 2
> step 3
> stopped after 3 steps
> ```
>
> The counter goes up after every step, so when the loop ends n is one more than the number of steps.

4. In the patrol, why does `patrol(stop_at)` return the number of steps rather than print it? [1 mark]

- ☐ **A** go_home needs the value to drive back the same distance
- ☐ **B** Printing is not allowed in functions
- ☐ **C** return is faster
- ☐ **D** So nothing appears on screen

> **Answer:** A. Return a value when another part of the program needs to use it.

5. What does this program print? [1 mark]

```
def patrol(readings):
    n = 0
    for d in readings:
        if d <= 30:
            break
        n = n + 1
    return n

print(patrol([60, 50, 40, 30, 20]))
```

> **Answer:**
>
> ```
> 3
> ```
>
> It counts steps until a reading is 30 or less: 60, 50 and 40 count, then it stops.

The task: patrol

Build the patrol against the wall: drive in 10 cm steps while the wall is more than 30 cm away, printing `step <n>: <distance> cm` each time; then print `arrived`, turn the LED green, and be inside the green zone.

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

def step(n):
    forward(50, distance=10)
    print(f"step {n}: {distance()} cm")

# the loop, then the ending
```

The hint students can ask for: Write one function for a single step that drives a little and reports where the wall is. Call it from a loop that keeps going while the wall is still far enough away, then finish with the arrival message and the LED.

A solution

```
from bugbot import *
connect()
def step(n):
    forward(50, distance=10)
    print(f'step {n}: {distance()} cm')
n = 1
while distance() > 30:
    step(n)
    n = n + 1
print('arrived')
led('green')
```

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