ANSWERS

1.2 Driving by distance

Driving · Robot club · about 15 min

What this lesson is about

Tell the robot how far, not how long. Where it thinks it is.

- distance=
- How it knows
- The robot coasts
- Distance and time together

Questions

7 marks in all

1. What changes when you add `distance=20` to `forward(60)`? [1 mark]

Tick every answer that is true.

- ☐ **A** The robot stops itself when it has gone 20 cm
- ☐ **B** Your program waits until the move is done
- ☐ **C** The robot drives for 20 seconds
- ☐ **D** The robot drives at speed 20

> **Answer:** A, B. With a distance, the robot stops on its own and nothing after the line runs until the move has finished.

2. How does the robot know how far it has gone? [1 mark]

- ☐ **A** An optical-flow sensor underneath watches the mat slide past, like a computer mouse
- ☐ **B** It counts how many times its wheels turn
- ☐ **C** It times how long it has been driving
- ☐ **D** A camera on the front measures the mat

> **Answer:** A. BugBot has no wheels to count. The sensor underneath works like the one in a mouse and adds up how far the mat has moved.

3. The robot starts at (0, 0) facing forward, drives forward 30 cm, then backward 10 cm. Roughly what does `position()` say? [1 mark]

- ☐ **A** (0, 20)
- ☐ **B** (20, 0)
- ☐ **C** (0, 40)
- ☐ **D** (40, 0)

> **Answer:** A. `x` is to the right and `y` is forward. 30 forward and 10 back leaves y at about 20, with x still about 0.

4. When is `driving on, not waiting` printed? [1 mark]

```
print("starting")
forward(60, distance=25)
print("arrived")
forward(60)
print("driving on, not waiting")
wait(1)
stop()
```

- ☐ **A** As soon as the second drive starts, while the robot is still moving
- ☐ **B** After the robot has gone 25 cm more
- ☐ **C** After the one second wait
- ☐ **D** Before "arrived"

> **Answer:** A. `forward(60, distance=25)` blocks until the move is done, but `forward(60)` returns straight away, so the next print happens at once.

5. `forward(100, distance=20)` usually lands further from 20 than `forward(30, distance=20)`. Why? [1 mark]

- ☐ **A** A faster robot coasts further after the motors stop, and the robot cannot allow for it perfectly
- ☐ **B** The sensor cannot measure at high speed
- ☐ **C** Speed 100 is not allowed with a distance
- ☐ **D** The slow one drives for longer, so it is more tired

> **Answer:** A. The robot switches off early to allow for the slide, but not perfectly. Faster means more slide, so more to get wrong.

6. The robot drives `forward(60, distance=20)` then `forward(60, distance=40)`. Roughly how many centimetres forward of the start does it end? Give a whole number. [1 mark]

> **Answer:** 60 (accept within 1). Each move counts on from where the last one stopped, so 20 then 40 more is about 60 cm.

7. There is a line 30 cm ahead that the robot must stop before. Which is the safest plan? [1 mark]

- ☐ **A** `forward(40, distance=27)`
- ☐ **B** `forward(100, distance=30)`
- ☐ **C** `forward(100)` then `wait(3)` then `stop()`
- ☐ **D** `forward(40, distance=33)`

> **Answer:** A. Leave a margin for coasting, and a lower speed coasts less. Driving by time does not know how far the robot has gone at all.

The task: twenty, then forty

Drive 20 cm, print `after 20:` followed by `position()`, then drive 40 cm more and stop inside the green band 60 cm from the start.

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

# drive forward at 60 for 20 cm, then stop
forward(60, distance=20)
print("after 20:", position())
```

The hint students can ask for: Drive 20 cm, print `after 20:` followed by position(), then drive 40 more and stop in the green band (60 cm from the start).

A solution

```
from bugbot import *
connect()
forward(60, distance=20)
print('after 20:', position())
forward(60, distance=40)
```

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