ANSWERS

1.1 Your first move

Driving · Robot club · about 12 min

What this lesson is about

Drive, wait, stop, the LED, and the safety rule.

- Drive, wait, stop
- How BugBot moves
- The safety rule
- Backwards

Questions

7 marks in all

1. In this program, what keeps the robot driving for two seconds? [1 mark]

```
led("green")
forward(50)
wait(2)
stop()
led("off")
```

- ☐ **A** `wait(2)`, because forward(50) returns straight away
- ☐ **B** `forward(50)`, because it drives for two seconds
- ☐ **C** `led("green")`
- ☐ **D** `stop()`

> **Answer:** A. `forward(50)` starts the robot and returns at once. The robot carries on while `wait(2)` lets two seconds pass.

2. What is the motion deadman? [1 mark]

- ☐ **A** The robot stops itself if it hears nothing from the program for half a second
- ☐ **B** A button that switches the robot off
- ☐ **C** The robot stops when it touches a wall
- ☐ **D** A command you must call at the end of every program

> **Answer:** A. It is a safety rule built into the robot. A crashed program cannot leave the robot driving into a wall.

3. On the real robot, a program is just `forward(50)` then `wait(2)`, with no `stop()`. About how many seconds after it starts do the motors switch off? Answer to one decimal place. [1 mark]

> **Answer:** 2.5 (accept within 0.2). It drives for the two seconds of the wait, then the program ends, and half a second later the deadman switches the motors off: about 2.5 seconds. The simulator switches them off as soon as the program ends.

4. Which of these are true about how BugBot moves? [1 mark]

Tick every answer that is true.

- ☐ **A** It has no wheels
- ☐ **B** Four small motors vibrate its feet
- ☐ **C** It can slide sideways without turning first
- ☐ **D** It drives perfectly straight
- ☐ **E** Every BugBot moves exactly the same

> **Answer:** A, B, C. Vibration slides the robot across the mat, which lets it go any direction. It also means no two BugBots move quite the same, and none goes perfectly straight.

5. Two programs both wait 2 seconds. One uses `forward(20)`, the other `forward(90)`. What is different? [1 mark]

- ☐ **A** The forward(90) one goes much further
- ☐ **B** Nothing, they both drive for 2 seconds
- ☐ **C** The forward(20) one goes further
- ☐ **D** The forward(90) one drives for longer

> **Answer:** A. The number is the speed. The same time at a higher speed covers more of the mat.

6. Put these lines in order to drive forward for 1 second, then backward for 2 seconds, then stop. [1 mark]

Number the lines 1 to 7 to put them in the right order.

___ `connect()`
___ `backward(60)`
___ `from bugbot import *`
___ `wait(1)`
___ `wait(2)`
___ `stop()`
___ `forward(60)`

> **Answer:**
>
> ```
> from bugbot import *
> connect()
> forward(60)
> wait(1)
> backward(60)
> wait(2)
> stop()
> ```
>
> Each drive command is followed by the wait that says how long it lasts. The motors go off at the end.

7. The robot drives forward 1.5 s and backward 1.5 s at the same speed, but the trail curves a little. What is the cause? [1 mark]

- ☐ **A** A vibration drive never goes perfectly straight
- ☐ **B** A bug in the program
- ☐ **C** The wait is too short
- ☐ **D** backward() is broken

> **Answer:** A. That curve is what a vibration drive does, not a mistake in your code. Later lessons use the sensors to correct it.

The task: cross the line

Drive from the start into the green zone and stop inside it, using only `forward`, `wait` and `stop`. If it stops short, wait longer or go faster. The zone is big because on a real mat no two runs land in quite the same place.

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

# drive forward at 50 (keeps going until the next command)
forward(50)
# pause 2 s (the robot keeps doing what it was told)
wait(2)
# all motors off
stop()
```

The hint students can ask for: The green zone starts 40 cm ahead. Drive long enough, or fast enough, to get there and stop inside it.

A solution

```
from bugbot import *
connect()
forward(60)
wait(4)
stop()
```

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