



F2.8 Project: parking sensor

What this lesson is about

A while loop, decisions and the buzzer: beep faster as the wall gets closer, and park.

- The brief
- Plan
- Step 1: drive to the bay
- Step 2: a beep that depends on the distance
- Step 3: put them together
- Testing

Questions

4 marks in all

1. Why does the parking sensor use a while loop rather than a for loop? [1 mark]

- ☐ A The robot cannot know beforehand how many times round it will take
- ☐ B for loops cannot use the sensor
- ☐ C while loops are faster
- ☐ D for loops cannot contain if

Answer: A. It repeats until the wall is close: a condition, not a count.

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

```
d = 35
if d > 50:
    gap = 0.4
elif d > 30:
    gap = 0.25
else:
    gap = 0.1
print(gap)
```

Answer:

0.25

35 is not over 50, but it is over 30, so the second branch runs.

3. The robot stops slightly closer to the wall than BAY. Why? [1 mark]

- ☐ A It coasts a little after stop()
- ☐ B distance() is always wrong
- ☐ C BAY is a string
- ☐ D The loop runs one extra time on purpose

Answer: A. A moving robot does not stop instantly, which is why you test and adjust BAY and the speed.

4. Where must `beeps = 0` go?

[1 mark]

- ☐ A Before the loop
- ☐ B Inside the loop, before the beep
- ☐ C After the loop
- ☐ D It is not needed

Answer: A. A count starts once, before the loop, and goes up inside it.

The task: parking sensor

Drive to the wall, beeping faster as it gets closer, and stop in the green bay without touching the wall. Then play a long note, turn the LED red and print one line in exactly this shape (your numbers will differ):

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

BAY = 20
beeps = 0

while distance() > BAY:
    forward(30)
    wait(0.1)
stop()
```

The hint students can ask for: Each time round the loop: move a little, read the gap, and choose the pause between beeps from that gap. Count the beeps as you go, and do the stopping, the long tone and the report after the loop ends.

A solution

```
from bugbot import *
connect()
BAY = 20
beeps = 0

while distance() > BAY:
    forward(30)
    d = distance()
    if d > 50:
        gap = 0.4
    elif d > 30:
        gap = 0.25
    else:
        gap = 0.1
    tone(880, 0.05)
    wait(gap)
    beeps = beeps + 1
stop()
tone(880, 1)
led("red")
print("stopped", distance(), "cm from the wall after", beeps, "beeps")
```

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