ANSWERS

# F11.9 Project: secure the robot

Cyber security · GCSE · OCR J277 1.4.2, AQA 8525 3.6.3, Edexcel 1CP2 4.2.1 · about 25 min

## What this lesson is about

Authenticate with a password, decrypt an encrypted command, and act on it.

- The protocol
- Try the pieces
- Plan it first

## Questions

**5 marks in all**

**1.** Why does the robot authenticate before taking a command? [1 mark]

- ☐ **A** So only someone with the password can control it
- ☐ **B** To make it faster
- ☐ **C** To save battery
- ☐ **D** To encrypt the mat

> **Answer:** A. Authentication keeps out impostors.

**2.** Why is the command encrypted? [1 mark]

- ☐ **A** So a listener on the radio cannot understand it
- ☐ **B** To make it shorter
- ☐ **C** So it arrives faster
- ☐ **D** So the robot can ignore it

> **Answer:** A. The radio is a broadcast; encryption keeps the command private.

**3.** A listener hears the Base's encrypted reply, KTWBFWI 25. What can they read without the key? [1 mark]

- ☐ **A** Only the number 25, because the Caesar cipher shifts letters and leaves digits alone
- ☐ **B** The whole command
- ☐ **C** Nothing at all
- ☐ **D** The key

> **Answer:** A. KTWBFWI is FORWARD shifted by 5, but the digits are sent as they are. A cipher that only scrambles part of a message leaks the rest.

**4.** Authentication and encryption together protect against what? [1 mark]

- ☐ **A** Impostors giving commands and listeners reading them
- ☐ **B** Only power cuts
- ☐ **C** Only malware
- ☐ **D** Only slow networks

> **Answer:** A. Authentication stops impostors; encryption stops eavesdroppers.

**5.** If the wrong password is sent, what should the robot do? [1 mark]

- ☐ **A** Not move
- ☐ **B** Drive anyway
- ☐ **C** Send its password
- ☐ **D** Turn off the radio

> **Answer:** A. No valid authentication means no command is carried out.

## The task: secure the robot

Authenticate with the Base using the password `bugbot42`, decrypt its reply with a Caesar cipher (key 5), and carry out the `FORWARD <cm>` command to reach the green depot, without collisions. Do not send any command before the password, and do not type the distance: read it from the decrypted message. The `caesar` helper is provided.

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

def caesar(text, key):
    out = ""
    for ch in text:
        if ch.isalpha():
            base = ord("A")
            out = out + chr((ord(ch.upper()) - base + key) % 26 + base)
        else:
            out = out + ch
    return out

def ask(question, seconds=1.5):
    send(question)
    for tick in range(int(seconds * 10)):
        wait(0.1)
        for sender, text in messages():
            return text
    return None
```

**The hint students can ask for:** Send the password and wait for the reply, which is encrypted. Decrypt it with the key shifting the other way, split the command into its word and its number, and drive that far.

### A solution

```
from bugbot import *
connect()
def caesar(text, key):
    out = ""
    for ch in text:
        if ch.isalpha():
            base = ord("A")
            out = out + chr((ord(ch.upper()) - base + key) % 26 + base)
        else:
            out = out + ch
    return out

def ask(question, seconds=1.5):
    send(question)
    for tick in range(int(seconds * 10)):
        wait(0.1)
        for sender, text in messages():
            return text
    return None

reply = ask("bugbot42")
if reply is None:
    led(255, 0, 0)
```

```
else:
    command = caesar(reply, -5)
    word, cm = command.split()
    forward(50, distance=int(cm))
    led(0, 255, 0)
```

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