



## F10.5 Addresses and packets

Networks · GCSE · OCR J277 1.3.2, AQA 8525 3.5, Edexcel 1CP2 4.1.3 ·  
about 15 min

### What this lesson is about

MAC and IP addresses, IPv4 and IPv6, and packet switching.

- MAC addresses
- IP addresses
- Packet switching

### Questions

6 marks in all

1. Which of these is a valid IPv4 address? [1 mark]

- ☐ A 192.168.0.12
- ☐ B 192.168.300.1
- ☐ C 3C:61:05:1A:9F:02
- ☐ D 10.0.1

**Answer:** A. Four denary numbers, each 0 to 255.

2. What is a MAC address used for? [1 mark]

- ☐ A Identifying a network interface, to deliver data within a local network
- ☐ B Routing data between networks
- ☐ C Naming a website
- ☐ D Encrypting data

**Answer:** A. Routers use IP addresses; switches use MAC addresses.

3. How many bits is an IPv6 address? [1 mark]

**Answer:** 128. IPv4 is 32 bits; IPv6 is 128.

4. Why do packets have sequence numbers? [1 mark]

- ☐ A So they can be put back in order when they arrive
- ☐ B So routers know the route
- ☐ C To encrypt them
- ☐ D To make them smaller

**Answer:** A. Packets can take different routes and arrive out of order.

5. Which is in a packet header?

[1 mark]

- ☐ A The destination IP address
- ☐ B The whole file
- ☐ C The user's password
- ☐ D The web page's title

**Answer:** A. Also the source address, sequence number and number of packets.

6. What does this program print?

[1 mark]

```
packets = ["2/2:Bot", "1/2:Bug"]
packets.sort()
print("".join(p[4:] for p in packets))
```

**Answer:**

BugBot

Sorted into order, the pieces join to BugBot.

## The task: packets

---

Write `make_packets(message, size)`, which splits `message` into pieces of `size` characters and returns them as packets in the form `<number>/<total>:<piece>`. Make the packets for `message` with a size of 10, print each one, and send each by radio. The packets arrive in the order `[packets[2], packets[0], packets[1]]`: sort them back into order by their sequence numbers, and print `reassembled: <message>` and `matches: True` (or `False`).

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

message = "robot 7 at 25,40 battery 87%"
```

**The hint students can ask for:** Slice the message into equal pieces and label each one with its number and how many there are in total. To reassemble, sort by the number at the front of each packet, then join the pieces back together.

## A solution

```
from bugbot import *
connect()
message = "robot 7 at 25,40 battery 87%"

def make_packets(message, size):
    pieces = [message[i:i + size] for i in range(0, len(message), size)]
    return [f"{n + 1}/{len(pieces)}:{piece}" for n, piece in enumerate(pieces)]

packets = make_packets(message, 10)
for packet in packets:
    print(packet)
    send(packet)
arrived = [packets[2], packets[0], packets[1]]
arrived.sort(key=lambda packet: int(packet.split("/")[0]))
text = "".join(packet.split(":", 1)[1] for packet in arrived)
print("reassembled:", text)
print("matches:", text == message)
```

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