ANSWERS

# F13.8 Revision that works

Exam preparation · GCSE · about 15 min

## What this lesson is about

Retrieval practice, spacing and interleaving, and a planner that schedules them.

- Three methods worth your time
- What to stop doing
- A plan you will actually follow
- Use what is already here
- The night before

## Questions

**5 marks in all**

**1.** Which revision method has the strongest evidence behind it? [1 mark]

- ☐ **A** Retrieval practice: testing yourself
- ☐ **B** Re-reading your notes
- ☐ **C** Highlighting
- ☐ **D** Copying out pages

> **Answer:** A. Getting it out of memory strengthens it more than putting it in again.

**2.** What is spacing? [1 mark]

- ☐ **A** Coming back to a topic days later, again and again
- ☐ **B** Leaving gaps in your notes
- ☐ **C** Revising one topic for a whole day
- ☐ **D** Taking breaks in a session

> **Answer:** A. Forgetting a little and recovering it is what makes it stick.

**3.** Why does interleaving feel harder? [1 mark]

- ☐ **A** You must work out which method each question needs
- ☐ **B** It takes more hours
- ☐ **C** The topics are harder
- ☐ **D** There are more questions

> **Answer:** A. That is exactly what the exam asks you to do.

**4.** Re-reading notes feels effective. Why is it not? [1 mark]

- ☐ **A** Familiar is not the same as known
- ☐ **B** It takes too long
- ☐ **C** Notes are usually wrong
- ☐ **D** It only works for practical topics

> **Answer:** A. Close the book and write what you remember instead.

5. A topic is rated 2 for confidence. With sessions = 6 - confidence, how many sessions? [1 mark]

> **Answer:** 4. 6 - 2.

## The task: build a revision plan

For each topic in `topics` (a name and a confidence from 1 to 5), the number of sessions is `6 - confidence`. Print `<name>: <n> sessions` for each, from the least confident to the most. Then spread them over the days in `days`, one session a day in turn, starting with the least confident topic, and print `<day>: <topic>` for each day. At the end print `sessions planned: <n>`, the number of days you filled, and `needs the most: <name>`, the topic needing the most sessions.

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

# name, confidence 1 to 5
topics = [("Networks", 2), ("Algorithms", 4), ("Cyber security", 3), ("Data
representation", 1)]
days = ["Mon", "Tue", "Wed", "Thu", "Fri", "Sat", "Sun"]
```

**The hint students can ask for:** Sessions are six take the confidence. Sort the topics by confidence, least first, and print them. Then walk the days, taking the next topic that still has sessions left and going back to the start of the list when you reach the end.

### A solution

```
from bugbot import *
connect()
topics = [("Networks", 2), ("Algorithms", 4), ("Cyber security", 3), ("Data
representation", 1)]
days = ["Mon", "Tue", "Wed", "Thu", "Fri", "Sat", "Sun"]
order = sorted(topics, key=lambda t: t[1])
left = {}
for name, confidence in order:
    left[name] = 6 - confidence
    print(f"{name}: {left[name]} sessions")
planned = 0
i = 0
for day in days:
    for tries in range(len(order)):
        name = order[i][0]
        i = (i + 1) % len(order)
        if left[name] > 0:
            left[name] = left[name] - 1
            planned = planned + 1
            print(f"{day}: {name}")
            break
print("sessions planned:", planned)
print("needs the most:", order[0][0])
```

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