ANSWERS

F13.9 Project: the revision robot

Exam preparation · GCSE · OCR J277 2.2.1, AQA 8525 3.2.2, Edexcel 1CP2 6.1.3 · about 25 min

What this lesson is about

A robot that asks you questions, marks your answers and tells you what to revise.

- What it does
- The question bank
- Marking an answer
- Plan it first

Questions

5 marks in all

1. Why does the marker lower and strip both answers before comparing? [1 mark]

- ☐ **A** So capitals and stray spaces do not count as wrong
- ☐ **B** To make it faster
- ☐ **C** Because input() returns a number
- ☐ **D** To sort them

> **Answer:** A. " An IP Address " and "an ip address" are the same answer.

2. How does the program know which topic to tell you to revise? [1 mark]

- ☐ **A** It counts the wrong answers per topic and names the highest
- ☐ **B** It picks the first topic
- ☐ **C** It asks you
- ☐ **D** It uses the longest question

> **Answer:** A. A dictionary of topic to count, then the largest.

3. What kind of revision is this program? [1 mark]

- ☐ **A** Retrieval practice
- ☐ **B** Re-reading
- ☐ **C** Highlighting
- ☐ **D** Copying

> **Answer:** A. You have to get the answer out of your memory.

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

```
def same(a, b):
    return a.strip().lower() == b.strip().lower()
print(same(' SMTP ', 'smtp'), same('IMAP', 'smtp'))
```

> **Answer:**
>
> ```
> True False
> ```
>
> The first pair match once cleaned; the second do not.

5. Asking the wrong ones again at the end of the quiz is an example of what? [1 mark]

- ☐ **A** More retrieval practice on the questions you find hardest
- ☐ **B** Spacing, because it comes days later
- ☐ **C** Cramming a whole topic
- ☐ **D** Highlighting

> **Answer:** A. It comes a few minutes later, not days, so it is not spacing: it is another go at getting back what you missed. Running the quiz again tomorrow would be spacing.

The task: the revision robot

Ask every question in `BANK` in turn. For each one, print `[<topic>] <question>`, read the answer with `input()`, and mark it ignoring capitals and spaces at the ends. Print `right` for a correct answer, or `wrong, it is <answer>`. Play a 880 Hz note for right and 220 Hz for wrong, each 0.2 seconds. At the end print `score: <n> of <total>` and `revise: <topic>`, the topic with the most wrong answers, then set the LED green if you scored more than half and red if not. The answers are typed for you.

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

BANK = [
    {"topic": "Networks", "q": "What does DNS turn a domain name into?", "a": "an IP
address"},
    {"topic": "Data", "q": "How many bits in a byte?", "a": "8"},
    {"topic": "Networks", "q": "Which protocol sends email?", "a": "SMTP"},
    {"topic": "Security", "q": "What does a firewall check?", "a": "traffic"},
]
```

The hint students can ask for: For each record, print the topic and question, read the answer, and compare both sides lowered and stripped. Count the score, and count the wrong answers per topic in a dictionary so you can name the worst at the end.

A solution

```
from bugbot import *
connect()
BANK = [
    {"topic": "Networks", "q": "What does DNS turn a domain name into?", "a": "an IP
address"},
    {"topic": "Data", "q": "How many bits in a byte?", "a": "8"},
    {"topic": "Networks", "q": "Which protocol sends email?", "a": "SMTP"},
    {"topic": "Security", "q": "What does a firewall check?", "a": "traffic"},
]

score = 0
wrong = {}
for item in BANK:
    print(f"[{item['topic']}] {item['q']}")
    given = input()
    if given.strip().lower() == item["a"].strip().lower():
        score = score + 1
        print("right")
        tone(880, 0.2)
    else:
        print(f"wrong, it is {item['a']}")
        tone(220, 0.2)
        wrong[item["topic"]] = wrong.get(item["topic"], 0) + 1
print(f"score: {score} of {len(BANK)}")
worst = ""
most = 0
for topic, n in wrong.items():
    if n > most:
        most = n
        worst = topic
print("revise:", worst)
```

```
if score > len(BANK) / 2:
    led(0, 255, 0)
else:
    led(255, 0, 0)
```

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