



F2.4 Nested selection and match

Decisions and loops · GCSE · OCR J277 2.2.1, AQA 8525 3.2.2, Edexcel
1CP2 6.2.2 · about 15 min

What this lesson is about

Decisions inside decisions, and choosing by exact value with a case statement.

- A decision inside a decision
- Tracing a nested decision
- Nested or joined?
- match: choosing by exact value

Questions

6 marks in all

1. What does this program print?

[1 mark]

```
d = 60
b = 10
if d < 30:
    if b < 20:
        print("stuck and flat")
    else:
        print("stuck")
else:
    if b < 20:
        print("free but flat")
    else:
        print("free")
```

Answer:

free but flat

d < 30 is False, so the outer else runs. Inside it, b < 20 is True.

2. Which else does an else belong to in nested selection?

[1 mark]

- ☐ A The if it lines up with
- ☐ B Always the first if in the program
- ☐ C Always the nearest if above it, whatever the indentation
- ☐ D The last if in the program

Answer: A. Indentation decides: an else pairs with the if at the same indentation.

3. What does this program print?

[1 mark]

```
command = "left"
match command:
    case "forward":
        print("f")
    case "left" | "right":
        print("sideways")
    case _:
        print("unknown")
```

Answer:

sideways

The first case that matches runs. | means or, so "left" matches the second case.

4. What does `case _:` do in a match?

[1 mark]

- ☐ A Matches any value not matched above, like else
- ☐ B Matches an empty string
- ☐ C Ends the match
- ☐ D Causes an error

Answer: A. The underscore matches anything, so it catches every other value. Put it last.

5. OCR's Exam Reference Language writes a case statement with which keyword? (one word)

[1 mark]

Answer: switch. switch ... case ... default ... endswitch. Python's version is match.

6. Which is the same as `if a > 30: if b > 50: print("go")` (with no else parts)?

[1 mark]

- ☐ A `if a > 30 and b > 50: print("go")`
- ☐ B `if a > 30 or b > 50: print("go")`
- ☐ C `if not a > 30: print("go")`
- ☐ D `if a > 30: print("go")`

Answer: A. Both conditions must be True to reach the print, which is exactly what and says.

The task: command mode

The robot asks `Which way?` and then `How far?`. Use `match` (or `elif`) on the direction to drive that far: `forward`, `back`, `left` or `right`. For any other word, print `unknown` and do not move. The task answers `right` and `20`; your program must work for any answers, so do not type them in.

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

command = input("Which way? ")
far = float(input("How far? "))
```

The hint students can ask for: Compare the command word against each option in turn, and give the case that matches nothing something sensible to do.

A solution

```
from bugbot import *
connect()
command = input("Which way? ")
far = float(input("How far? "))
match command:
    case "forward":
        forward(50, distance=far)
    case "back":
        backward(50, distance=far)
    case "left":
        left(50, distance=far)
    case "right":
        right(50, distance=far)
    case _:
        print("unknown")
```

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