



F3.5 Two-dimensional arrays

Strings, lists and records · GCSE · OCR J277 2.2.3, AQA 8525 3.2.6, Edexcel
1CP2 6.3.1 · about 15 min

What this lesson is about

Rows and columns: the depth grid as a list of lists.

- A list of lists
- Visiting every item
- The depth grid
- Finding something in the grid

Questions

6 marks in all

1. What does this program print?

[1 mark]

```
board = [[1, 2, 3], [4, 5, 6], [7, 8, 9]]  
print(board[1][2])
```

Answer:

6

Row 1 is [4, 5, 6], and column 2 of it is 6.

2. What does this program print?

[1 mark]

```
board = [[1, 2], [3, 4]]  
total = 0  
for row in board:  
    for item in row:  
        total = total + item  
print(total)
```

Answer:

10

The two loops visit every item: 1 + 2 + 3 + 4.

3. A 2D array has 8 rows and 8 columns. How many items does it hold?

[1 mark]

Answer: 64. Rows times columns.

4. The depth grid comes as one flat list of 64 readings, row after row. Which slice is row 2?

[1 mark]

- ☐ A flat[16:24]
☐ B flat[2:10]
☐ C flat[8:16]
☐ D flat[16:32]

Answer: A. Row r starts at $r * 8$ and has 8 items, so row 2 is from 16 up to 24.

5. What does this program print?

[1 mark]

```
grid = [[0, 0], [0, 0]]
grid[1][0] = 5
print(grid)
```

Answer:

```
[[0, 0], [5, 0]]
```

grid[1][0] is row 1, column 0.

6. In OCR's Exam Reference Language, how is row 3, column 4 of `grid` written?

[1 mark]

- ☐ A `grid[3, 4]`
- ☐ B `grid[3][4]`
- ☐ C `grid(3, 4)`
- ☐ D `grid.3.4`

Answer: A. The Reference Language puts both indexes in one pair of brackets. Python and AQA use two.

The task: nearest column

There is a box off to one side. Without moving, find the smallest reading in the level rows (2 and 3) of the depth grid and print `nearest: <cm> cm in column <n>`.

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

flat = tof_grid()
best = min(flat)
print("nearest:", best)
```

The hint students can ask for: The grid is rows of readings. Look along the row you care about, keeping both the smallest reading and the position it was found at.

A solution

```
from bugbot import *
connect()
grid = tof_grid()
best = 999
best_col = 0
for row in (2, 3):                                # the level rows; the rows below see the mat
    for col in range(8):
        d = grid[row * 8 + col]
        if d < best:
            best = d
            best_col = col
print(f'nearest: {best} cm in column {best_col}')
```

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

