



F5.4 Trace tables

Algorithms · GCSE · OCR J277 2.1.2, AQA 8525 3.1.1, Edexcel 1CP2 1.2.4 ·
about 15 min

Name _____

Class _____

Date _____

What this lesson is about

Following an algorithm line by line, and tracing to find a logic error.

- A first trace
- Let the program trace itself
- Tracing to find a bug
- Tracing a robot

Questions

5 marks in all

1. What does this program print?

[1 mark]

```
total = 0
for i in range(1, 5):
    total = total + i * 2
print(total)
```

2. What does this program print?

[1 mark]

```
x = 1
y = 10
while x < y:
    x = x * 2
    y = y - 1
print(x, y)
```

3. A program prints the average of five readings as 4.0 when it should be 30.0. A trace shows total goes 40, 35, 30, 25, 20. What is the bug? [1 mark]

- ☐ A total = 0 is inside the loop, so it resets each time
- ☐ B The readings are wrong
- ☐ C The loop runs too many times
- ☐ D Division is broken

4. What is a trace table used for?

[1 mark]

- ☐ A Recording variable values step by step to check what an algorithm does
- ☐ B Storing a program's data
- ☐ C Drawing a flowchart
- ☐ D Timing a program

5. What does this program print?

[1 mark]

```
n = 20
steps = 0
while n > 1:
    n = n // 2
    steps = steps + 1
print(n, steps)
```

The task: trace and fix

Fix the average program so it prints `average: 30.0`. Keep a trace: inside the loop, print a line `i=<i> total=<total>` each time round, so the program shows its own trace table.

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

readings = [40, 35, 30, 25, 20]
for i in range(len(readings)):
    total = 0
    total = total + readings[i]
print("average:", total / len(readings))
```

Plan your program here, then type it in and press Run.



Do it on the robot

www.bugbotlab.com/learn/f5-4-trace-tables/

The simulator checks it and tells you when it passes. Nothing to install, no account.

Challenges

1. Trace `n = 27` then `while n != 1: if n is even, n = n // 2, else n = 3 * n + 1` for the first six rows. Then run it to see how long it really takes.
2. Write a trace table for the parking sensor's loop, with the distance, the gap and the beep count.
3. Swap the two lines inside the `x` and `y` loop above. Trace it again before running. Does it still stop?