



F3.4 Iterating over a list

Strings, lists and records · GCSE · OCR J277 2.2.3, AQA 8525 3.2.6, Edexcel

1CP2 6.2.2 · about 15 min

Name _____

Class _____

Date _____

What this lesson is about

Loops over items and indexes, totals, counts, min and max, and linear search.

- for over the items
- for over the indexes
- The patterns, over a list
- Readings from the robot
- Searching a list

Questions

6 marks in all

1. What does this program print?

[1 mark]

```
total = 0
for r in [4, 7, 9]:
    total = total + r
print(total / 3)
```

2. What does this program print?

[1 mark]

```
colours = ["red", "green", "blue"]
for i in range(len(colours)):
    print(i, colours[i])
```

3. What does this program print?

[1 mark]

```
readings = [41, 35, 30, 25]
count = 0
for r in readings:
    if r < 32:
        count = count + 1
print(count)
```

4. A linear search finds its target at index 2. Why does it use `break` ?

[1 mark]

- ☐ A There is no need to look at the rest of the list
- ☐ B To restart the search
- ☐ C `break` is needed in every loop
- ☐ D To print the index

5. How does a linear search work?

[1 mark]

- ☐ A It checks each item in turn, from the start, until it finds the target or runs out
- ☐ B It jumps to the middle and halves the list
- ☐ C It sorts the list first
- ☐ D It only checks the first and last items

6. What does this program print?

[1 mark]

```
readings = [40, 36, 25]
for i in range(1, len(readings)):
    print(readings[i - 1] - readings[i])
```

The task: sensor log

Drive forward in five steps of 8 cm, recording `distance()` before each step in a list. Then print four lines: `readings: <the list>`, `smallest: <cm>`, `largest: <cm>` and `average: <cm>`. Work the average out with a loop or with `sum` and `len`, not by typing numbers.

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

readings = []
```

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



Do it on the robot

www.bugbotlab.com/learn/f3-4-iterating-over-a-list/

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

Challenges

- Find the smallest reading with a loop, without using `min`.
- Drive a shape from a list of `[distance, turn]` pairs, such as `[[30, 90], [20, 90], [30, 90], [20, 90]]`.
- Count how many names in a list start with the letter `A`.