



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

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)
```

Answer:

6.666666666666667

The running total is 20, and the average is 20 divided by 3.

2. What does this program print?

[1 mark]

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

Answer:

0 red
1 green
2 blue

range(len(colours)) gives every valid index, 0 to 2.

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)
```

Answer:

2

Only 30 and 25 are under 32.

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

Answer: A. Once the item is found, looking further only wastes time.

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

Answer: A. Linear search looks at the items one by one. It works on any list, sorted or not.

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])
```

Answer:

4
11

Starting at index 1 means each item can be compared with the one before it.

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 = []
```

The hint students can ask for: Take a reading, move a short way, and repeat, adding each reading to a list as you go. The smallest, largest and average all come from that list afterwards.

A solution

```
from bugbot import *
connect()
readings = []
for i in range(5):
    readings.append(distance())
    forward(50, distance=8)
print("readings:", readings)
print("smallest:", min(readings))
print("largest:", max(readings))
print("average:", sum(readings) / len(readings))
```

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