ANSWERS

F3.8 Project: survey the room

BugBotLab

Strings, lists and records · GCSE · about 25 min

What this lesson is about

Look in eight directions, store the survey as records, report on it and head for open space.

- The brief
- Plan the data first
- Step 1: collect the survey
- Step 2: a table
- Step 3: the report
- Test with data you can check

Questions

4 marks in all

1. Why plan what one record looks like before writing the survey loop? [1 mark]

- ☐ **A** Every later loop depends on the fields the records have
- ☐ **B** Python needs records declared first
- ☐ **C** It makes the robot turn faster
- ☐ **D** Records cannot be changed later

> **Answer:** A. The report loops all read fields, so deciding the fields first keeps the whole program consistent.

2. What does this program print? [1 mark]

```
survey = [{"direction": 0, "distance": 31.0}, {"direction": 45, "distance": 42.5},
{"direction": 90, "distance": 51.0}]
total = 0
for r in survey:
    total = total + r["distance"]
print(total / len(survey))
```

> **Answer:**
>
> ```
> 41.5
> ```
>
> The distances total 124.5, and divided by 3 records that is 41.5.

3. What does this program print? [1 mark]

```
survey = [{"direction": 0, "distance": 31.0}, {"direction": 45, "distance": 42.5},
{"direction": 90, "distance": 51.0}]
open_count = 0
for r in survey:
    if r["distance"] > 40:
        open_count = open_count + 1
print(open_count)
```

> **Answer:**
>
> ```
> 2
> ```
>
> 42.5 and 51.0 are over 40.

4. Why test the report on a small made-up survey before using the robot's data? [1 mark]

- ☐ **A** You can work the right answers out by hand and compare
- ☐ **B** Made-up data is more accurate
- ☐ **C** The robot cannot store records
- ☐ **D** It is required by Python

> **Answer:** A. Test data with known answers shows whether the program is right.

The task: survey the room

Do the survey from the brief. Print the table, then report lines in exactly this shape (these are the numbers a correct program gets in this room): Then turn to face the farthest direction and play a note.

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

survey = []
for i in range(8):
    turn_right(30, angle=45)
```

The hint students can ask for: Build the list of records first, then answer each question with its own pass over that list. Turn to the direction stored in the record you chose.

A solution

```
from bugbot import *
connect()
survey = []
for i in range(8):
    survey.append({"direction": i * 45, "distance": distance()})
    turn_right(30, angle=45)

print("direction  distance")
for r in survey:
    print(f"{r['direction']:9}  {r['distance']:8}")

nearest = survey[0]
farthest = survey[0]
total = 0
open_count = 0
for r in survey:
    if r["distance"] < nearest["distance"]:
        nearest = r
    if r["distance"] > farthest["distance"]:
        farthest = r
    total = total + r["distance"]
    if r["distance"] > 40:
        open_count = open_count + 1

print(f"nearest: {nearest['direction']} degrees, {nearest['distance']} cm")
print(f"farthest: {farthest['direction']} degrees, {farthest['distance']} cm")
print(f"average: {round(total / len(survey), 1)} cm")
print("open directions:", open_count)
turn_right(30, angle=farthest["direction"])
tone(880, 0.5)
```

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