



F5.10 Project: sort the readings

What this lesson is about

Survey the room, sort the records, answer questions from sorted data, and choose the algorithms.

- The brief
- Plan: which algorithms?
- Step 1: the survey
- Step 2: sort records, not numbers
- Step 3: answer questions from sorted data
- Testing

Questions

4 marks in all

1. After sorting survey records nearest first, where is the farthest direction? [1 mark]

- ☐ A In the last record
- ☐ B In the first record
- ☐ C In the middle record
- ☐ D It must be searched for

Answer: A. Sorting puts the largest distance at the end.

2. The sorted distances are 15, 16, 16, 26, 26, 31, 31, 34, 34, 35, 51, 57. What is the median? [1 mark]

Answer: 31. With 12 values the median is the average of the 6th and 7th: 31 and 31.

3. To sort records by distance, what does the sort compare? [1 mark]

- ☐ A The distance field of each record
- ☐ B The whole records
- ☐ C The direction field
- ☐ D The position in the list

Answer: A. Compare one field, and move whole records so each distance keeps its direction.

4. Why is any of the three sorts fine for 12 readings? [1 mark]

- ☐ A With so few items, the difference in speed is too small to matter
- ☐ B They all do exactly the same number of comparisons
- ☐ C Only merge sort works on records
- ☐ D Twelve is an even number

Answer: A. Efficiency matters for large data; for 12 items, choose what you can write and explain correctly.

The task: sort the readings

Do the survey from the brief. Sort the records **nearest first** with a sorting algorithm you write (no `sort` or `sorted`). Print one line `sorted:` followed by the distances in order, then `median:` `<cm>` `cm` and `farthest:` `<degrees>` `degrees`. Turn to face the farthest direction and beep.

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

survey = []
for i in range(12):
    turn_right(30, angle=30)
```

The hint students can ask for: Survey into a list of records first, then sort those records by their distance. The middle of an even-length list is the average of the two middle values, and the last one is the largest.

A solution

```
from bugbot import *
connect()
survey = []
for i in range(12):
    survey.append({"direction": i * 30, "distance": distance()})
    turn_right(30, angle=30)

swapped = True
while swapped:
    swapped = False
    for i in range(len(survey) - 1):
        if survey[i]["distance"] > survey[i + 1]["distance"]:
            survey[i], survey[i + 1] = survey[i + 1], survey[i]
            swapped = True

distances = []
for r in survey:
    distances.append(r["distance"])
print("sorted:", distances)
print("median:", (survey[5]["distance"] + survey[6]["distance"]) / 2, "cm")
print("farthest:", survey[-1]["direction"], "degrees")
turn_right(30, angle=survey[-1]["direction"])
tone(880, 0.4)
```

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