



F13.4 Programming questions

Exam preparation · GCSE · about 20 min

Name _____

Class _____

Date _____

What this lesson is about

Taking a question apart, the jobs that come up again and again, and how the marks are given.

- Read the question like a specification
- The jobs that come up again and again
- How the marks are given
- Worked example

Questions

5 marks in all

1. You cannot finish a programming question. What is the best thing to do? [1 mark]
 - ☐ A Write what is left as comments or pseudocode
 - ☐ B Leave it blank
 - ☐ C Rub out what you have
 - ☐ D Copy an earlier answer
2. Which of these earns marks in a written programming answer? [1 mark]
 - ☐ A Meaningful variable names and clear structure
 - ☐ B Very short variable names
 - ☐ C No comments at all
 - ☐ D Perfect punctuation
3. A question says to reject an input over 60. What is that worth? [1 mark]
 - ☐ A A mark: handle the failure case
 - ☐ B Nothing
 - ☐ C Only if the program runs
 - ☐ D A mark only in an on-screen exam
4. What does this program print? [1 mark]

```
done = 0
steps = 0
while done < 32:
    step = min(7, 32 - done)
    done = done + step
    steps = steps + 1
print(steps, done)
```

5. What should you do before writing any code in a long question?

[1 mark]

- ☐ A Underline the inputs, outputs and rules
- ☐ B Write the last line first
- ☐ C Choose variable names for everything
- ☐ D Count the marks

The task: an exam-style program

Write the program for this question. The robot must drive `far` centimetres in steps of `step` centimetres, where the last step is whatever is left over. After each step, print `so far: <n> cm`. At the end print `steps: <n>` and `total: <n> cm`. Use the values given; do not type the answers.

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

far = 32
step = 7
```

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



Do it on the robot

www.bugbotlab.com/learn/f13-4-programming-questions/

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

Challenges

1. Change `far` to 30 and `step` to 5. Does your program still print the right number of steps?
2. Add validation: refuse a `step` of 0 and say why, instead of looping forever.
3. Rewrite it with a `for` loop instead of a `while` loop. Which reads better here?