

# Kit's Logic Lab: Mastering Algorithms

*Learning objective: To understand how to design, write, and debug programs by using logical reasoning and sequence.*

Read each question carefully. Use your knowledge of algorithms, loops, and debugging to solve these computing challenges. Remember: a program is only as good as the instructions you give it!

*Word bank: Algorithm · Debug · Sequence · Loop · Variable · Input · Output · Syntax*

**1. Kit the fox is writing an algorithm for a robot to make a jam sandwich. The robot has stopped working because the instructions are not in the correct order. Re-order these steps: (1) Spread jam on bread, (2) Get bread from the cupboard, (3) Put two slices together, (4) Open the jam jar. Explain why the order of these steps is critical for a successful program. (3 marks)**

---

---

---

**2. Define the term 'debug' in your own words. If Kit's computer program keeps crashing every time he tries to save a file, what process should he follow to find and fix the error? (2 marks)**

---

---

**3. Look at this piece of pseudocode: 'Repeat 4 times: [Move forward 10 steps, Turn right 90 degrees]'. What shape will this algorithm create? Justify your answer by describing the pattern of movement. (2 marks)**

---

---

**4. Compare and contrast a 'sequence' and a 'loop'. Why might a programmer choose to use a loop instead of writing out individual instructions multiple times? (3 marks)**

---

---

---

5. If you were programming a digital game where a character collects coins worth £0.50 each, how would you use a 'variable' to keep track of the player's total score? (2 marks)

---

---

6. What might happen if a computer program contains a syntax error? Explain the potential impact on the 'output' of the program. (2 marks)

---

---

**Draw:** Draw a flowchart that shows the logic for a simple task, such as brushing your teeth or getting ready for school. Use diamond shapes for decisions (Yes/No) and rectangles for actions.



*Extension challenge: Design a 'Greater Depth' challenge: Write a short algorithm using loops for a character to navigate a grid from (0,0) to (5,5) while avoiding obstacles. Explain why your chosen path is the most efficient.*