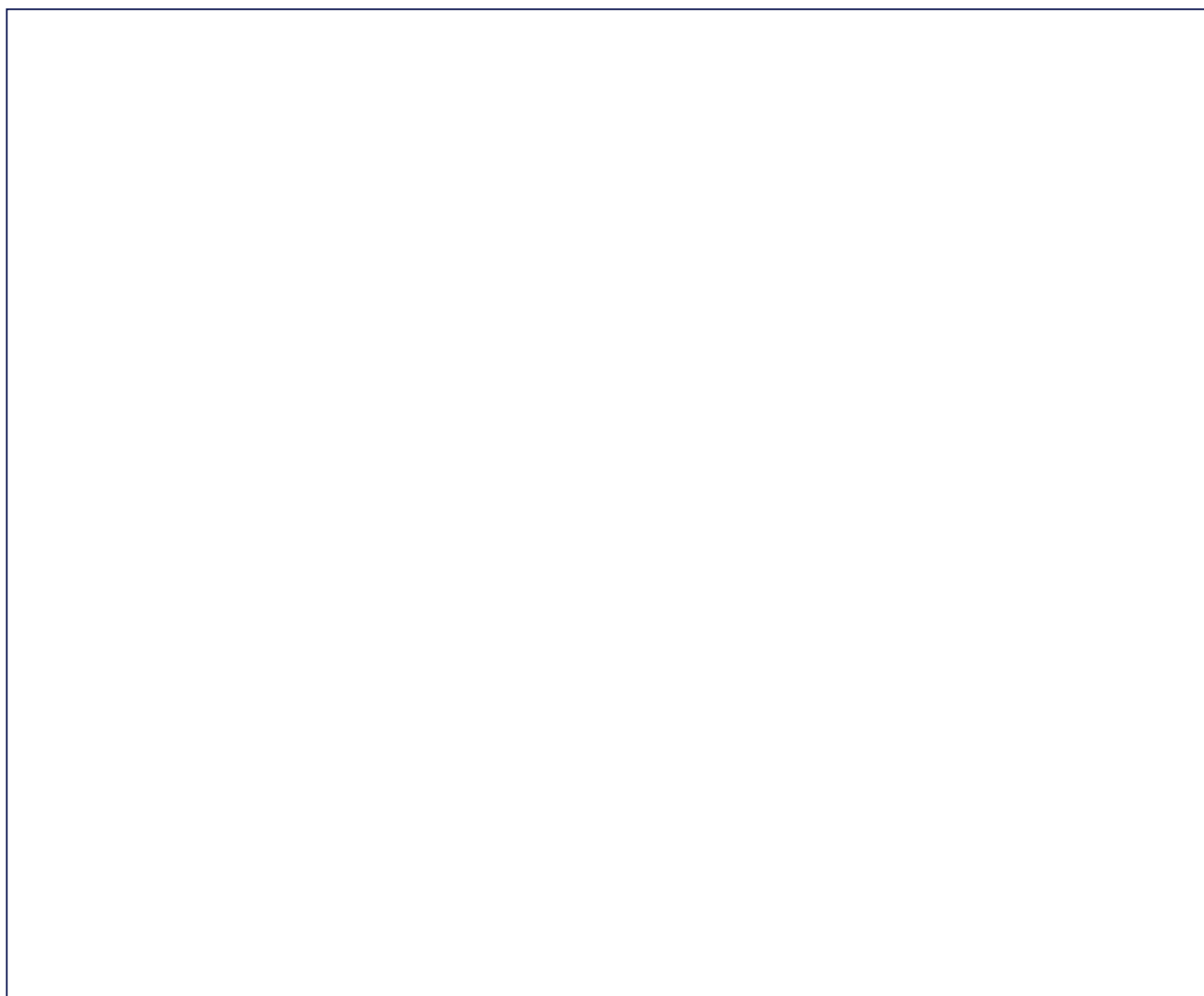


Kit's Computing Challenge: Designing an Algorithm

Learning objective: To design an algorithm for a sprite to navigate a maze and understand the importance of sequence and debugging.

Kit the fox is building a new digital game. Use the space below to draw a top-down view of a simple maze. Then, draw a path for a sprite to follow to reach the 'Finish' flag. Label the components of your code using the word bank provided.

Draw: Draw a top-down, clean schematic of a digital maze. Use a geometric, grid-based style. Include a small square 'sprite' character at the start. Use arrows to show the path taken through the maze. Ensure there is clear space around the drawing for labels from the word bank to be connected to specific parts of the diagram.



Label words: Algorithm · Sequence · Sprite · Loop · Bug · Debug · Input · Output

Steps:

1. Step 1: Draw a 5x5 grid. Place a 'Start' square at the top-left and a 'Finish' square at the bottom-right. Draw a path that avoids two 'obstacle' walls.
2. Step 2: Write a short sequence of instructions (e.g., 'Move forward 2 spaces') to get your sprite from start to finish.
3. Explain why the order of your instructions (the sequence) is vital for the sprite to reach the finish

successfully.

4. What might happen if your code contains a 'bug'? How would you go about debugging it?
5. Compare and contrast an algorithm written for a computer versus a set of instructions written for a human to bake a cake. How are they similar?
6. Justify why using a 'loop' is more efficient than repeating the same command multiple times in your code.