

Grade 3 Computing: Programming and Algorithms Assessment

Learning objective: To demonstrate understanding of algorithms, debugging, and the use of repetition in programming.

Read each question carefully. Use the word bank to help you express your ideas clearly. Show your working where necessary.

Kit the fox is designing a computer game where a robot navigates a maze to collect golden gears. To make the robot move, Kit must write a precise set of instructions. Sometimes the robot hits a wall, which means Kit has to find the error and fix it before the game can continue.

Word bank: algorithm · debug · sequence · loop · variable · syntax · input · output · repetition

1. Define the term 'algorithm' in your own words. Why is it important for the instructions to be in a specific order? (2 marks)

2. Kit uses a 'loop' in his code to make the robot walk forward five steps. Explain why using a loop is more efficient than writing 'move forward' five separate times. (2 marks)

3. If the robot stops moving halfway through the maze, what is the process of finding and fixing the mistake called? Describe one way you could check your code for errors. (2 marks)

4. Look at this sequence: 1. Move forward, 2. Turn left, 3. Move forward, 4. Turn left. If the robot performs this sequence three times, describe the shape the robot has traced on the floor. Justify your answer. (3 marks)

5. Compare and contrast a 'sequence' and a 'loop'. How do they work together to create a complex program? (3 marks)

6. Kit wants to add a feature where the robot collects a gear worth £5. If the robot collects 4 gears, write a simple algorithm or pseudo-code to calculate the total value of the collected gears. (3 marks)

7. Explain why it is vital for a programr to test their code multiple times. What might happen if a programr skips the testing phase before releasing their game? (5 marks)

Draw: Draw a flow chart showing the steps Kit's robot takes to move from a start point, avoid one obstacle, and reach a finish line.



Extension challenge: Imagine you are designing a program for a school kitchen. Write a detailed algorithm for a robot to prepare a piece of toast. Include a conditional statement (e.g., 'If the toast is burnt, then...').
Total: /25