

Kit's Code-Cracking Challenge

Learning objective: To understand algorithms, debugging, and sequencing in computer programming.

Help Kit the fox fix his buggy code! Read each question carefully and circle the best answer. Good luck, young programmers!

Answer Key: 1. A (Algorithm); 2. True; 3. C (To fix errors); 4. Sequence; 5. B (A loop); 6. False; 7. C (Computer program); 8. Debugging; 9. B (Order matters); 10. True. Note: Debugging is the essential process of locating and correcting errors in code.

Word bank: algorithm · debug · sequence · loop · input · variable · syntax · program

**1. 1. What is the correct term for a precise set of step-by-step instructions for a computer?
A) Algorithm* B) Recipe C) Script D) Pattern (1 mark)**

2. 2. True or False: If you swap the order of two instructions in a sequence, the outcome of the program might change. (1 mark)

3. 3. What does it mean to 'debug' a piece of code? A) Delete it B) Start over C) Find and fix errors* D) Add more colors (1 mark)

4. 4. What do we call the specific order in which instructions are performed? (1 word) (1 mark)

5. 5. Which command would you use to make an action repeat itself multiple times? A) A condition B) A loop* C) A variable D) A sequence (1 mark)

6. 6. True or False: Computers are clever enough to understand exactly what you mean, even if your instructions are out of order. (1 mark)

7. 7. A set of instructions that a computer follows is called a: A) Story B) Map C) Program* D) Puzzle (1 mark)

8. 8. If your robot stops moving halfway through a maze, what process must you perform to find the mistake? (1 word) (1 mark)

9. 9. Why is the 'sequence' of code important? A) It makes the code look pretty B) It determines the order of events* C) It saves memory D) It creates bugs (1 mark)

10. 10. True or False: An algorithm can be written for daily tasks, like making a cup of tea, as well as for computers. (1 mark)

Draw: Draw a flowchart showing the steps to 'brush your teeth'. Use boxes for actions and arrows to show the sequence.



Extension challenge: Greater Depth Challenges: 1. Explain why a program might run perfectly but still not achieve the intended result. 2. Compare and contrast a 'loop' with a single 'sequence' of instructions. 3. Justify why it is important to test code in small sections rather than all at once. 4. What might happen if an algorithm had a logical error in the middle of a loop? 5. Design a short algorithm to help a robot navigate a grid from (0,0) to (3,3). 6. If you were teaching a younger student about debugging, what analogy would you use to explain why mistakes are actually helpful?