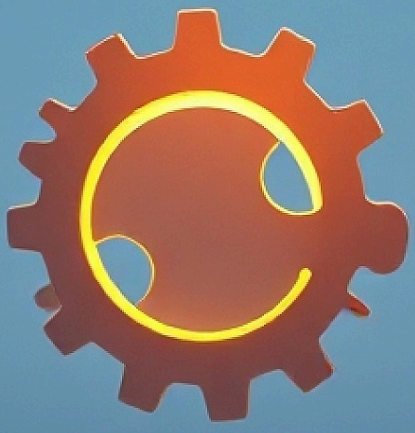
What is a logic error?



www.yzokul.com.tr

What is a logic error?



A logic error happens when a program runs and finishes but gives a wrong result, not the answer you expected. The program does not stop with an error message. It simply produces a number or output that is not correct. This is different from a program that crashes or refuses to run. A logic error is about getting the wrong answer even though the program seems to work.



A program simply does what its instructions say, step by step. It does not know what answer you wanted or what result is correct. It follows each line exactly as written, in order. If an instruction says to divide by 5, the program divides by 5, even if you meant to divide by 10. The computer has no idea about your intention; it only follows the plan you gave it.



A logic error means the program finishes without crashing. It runs from start to end and produces some output. The problem is only that the result is not the expected one. For example, a program that should print 85 might print 42. Nothing stops the program; it just gives a wrong answer. The mistake is in the logic, meaning the reasoning or plan behind the steps.



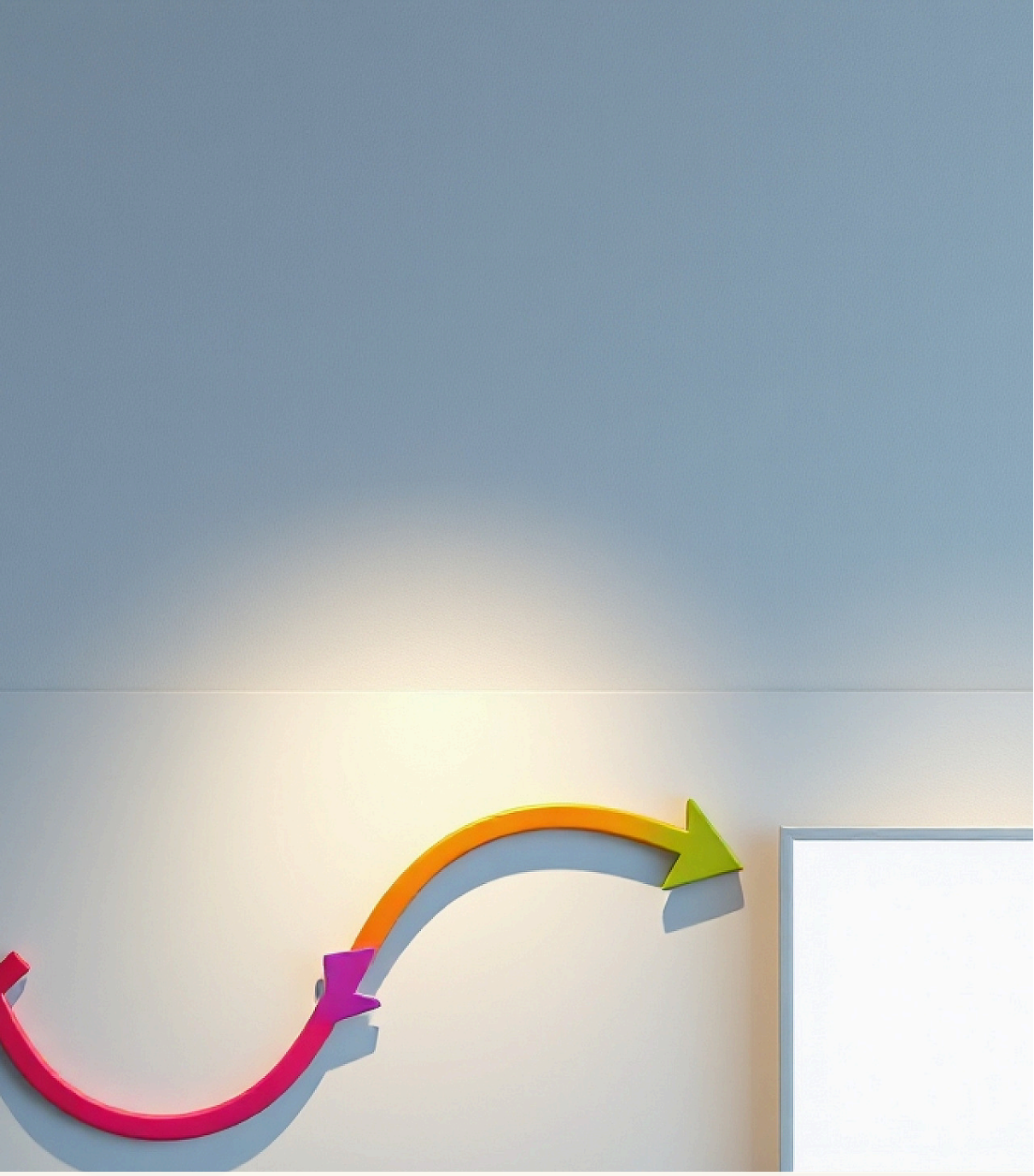
Look at a small program that calculates the average of students' scores. It adds the scores together into a total. Suppose there are three students with scores 80, 90, and 100. The program adds them: 80 + 90 + 100 = 270. So the total is 270. The next step should divide this total by the number of students, which is 3, to get the average.



The program divides the total by the wrong number instead of the number of students. For example, it might divide 270 by 2 instead of 3. The average it prints is therefore wrong: 270 divided by 2 is 135, but the correct average is 270 divided by 3, which is 90. The program ran without stopping, but the output is not the expected answer. This is a logic error.



The computer performed the division correctly. If you tell it to divide 270 by 2, it gives 135, which is the right answer to that division. The mistake is in how the solution was set up, not in the machine. The computer did its job; the instructions told it to do the wrong thing. A logic error is a mistake in the plan, not a broken computer.



A logic error comes from a wrong plan or a wrong step in the instructions. The computer follows the wrong plan faithfully. It does not question the instructions; it just executes them. So if the plan says to divide by the wrong number, the computer will do that and give a wrong result. The error is in the thinking behind the program, not in the computer's ability to calculate.



These mistakes often produce no clear error message. The program looks fine and simply gives a wrong answer. There is no red text, no warning, no crash. Everything seems normal. That is why logic errors can be hard to find for beginners. You might think the program is correct because it runs without complaining, but the output tells a different story.



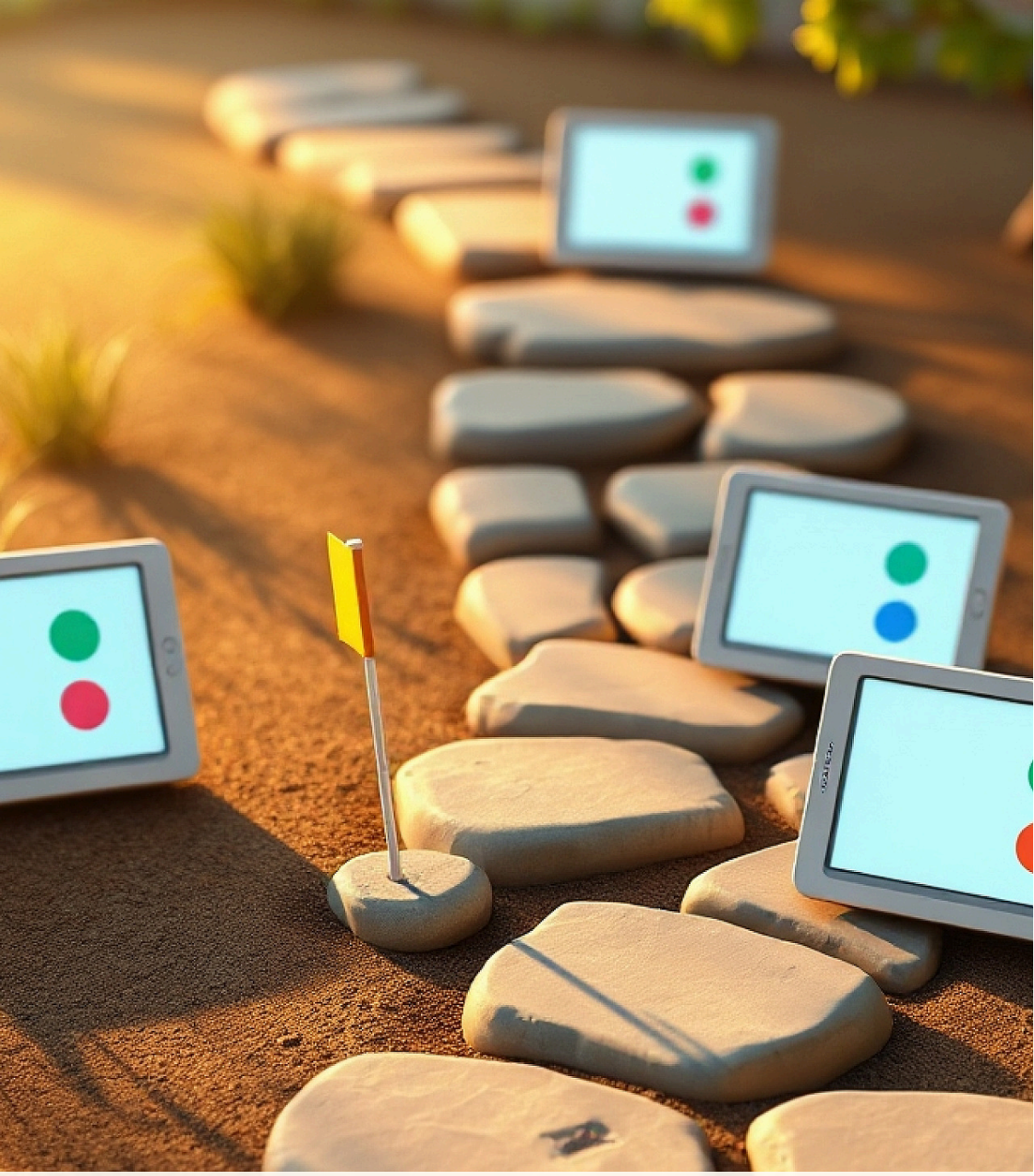
The wrong output can look like a normal number, so nothing warns you. For example, 135 is a perfectly normal number; it does not look strange. You have to check the result yourself to see that it is wrong. The program does not know it is wrong, and it does not tell you. You must compare the output with what you expected to find the logic error.



Try the program with a case where you already know the right answer. For example, use scores 80, 90, and 100, where the average should be 90. Run the program and see what it prints. Comparing the two shows whether the program is wrong. If it prints 135 instead of 90, you know there is a logic error. This comparison is a simple way to check.



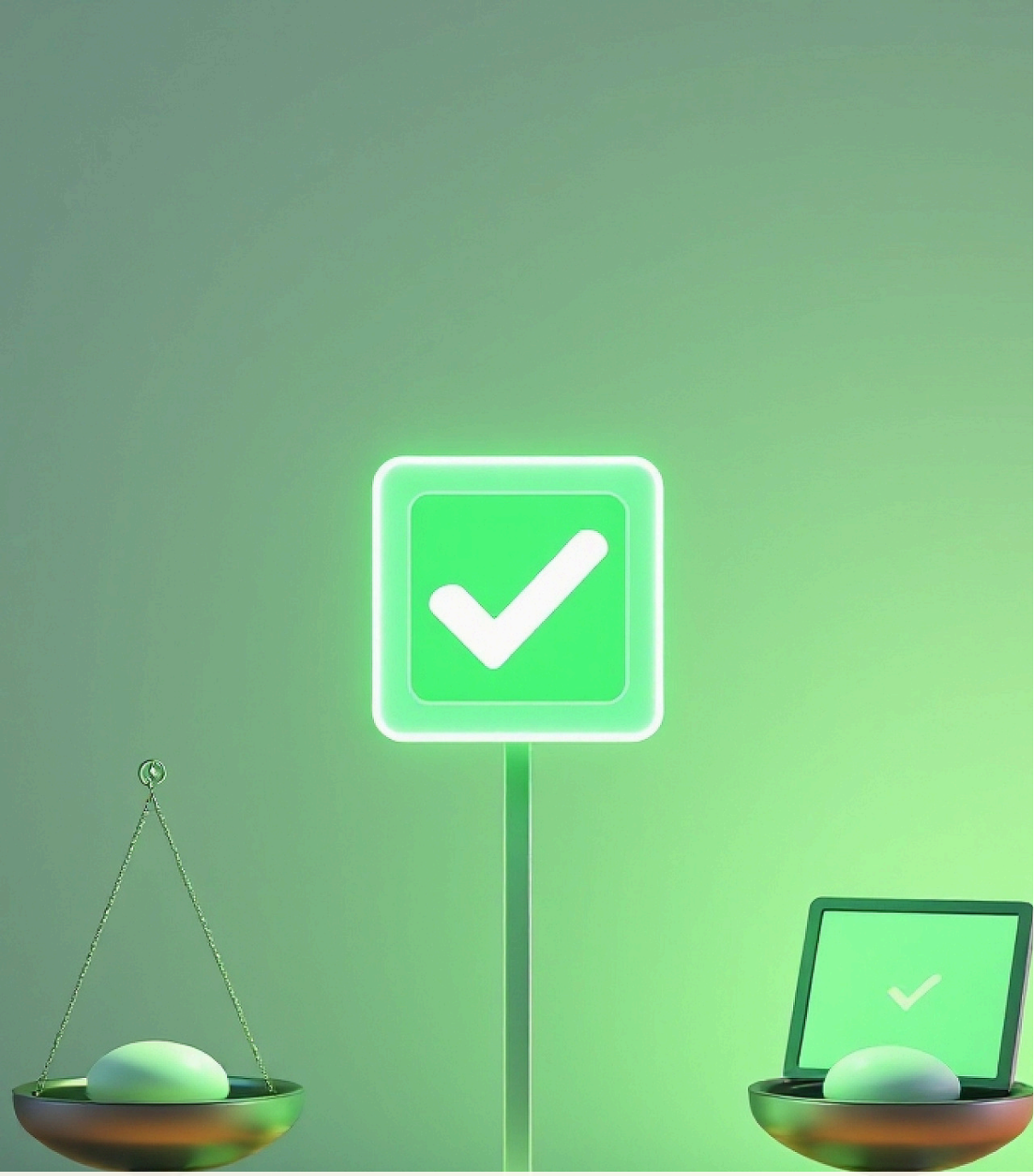
Run the program with several different inputs, not just one. Trying more examples helps the mistake show up clearly. For instance, try two students, then four students, and see if the average is always wrong in the same way. If the program divides by the wrong number, the error will appear in each example. More examples give you more evidence and help you find the pattern.



Trace the steps of the program to find where the plan goes wrong. Read each instruction and check if it matches what you want to happen. In the average example, the wrong step is the number used for dividing. The program divides by 2 instead of 3. Once you find that step, you have found the logic error. Tracing means following the program's steps one by one.



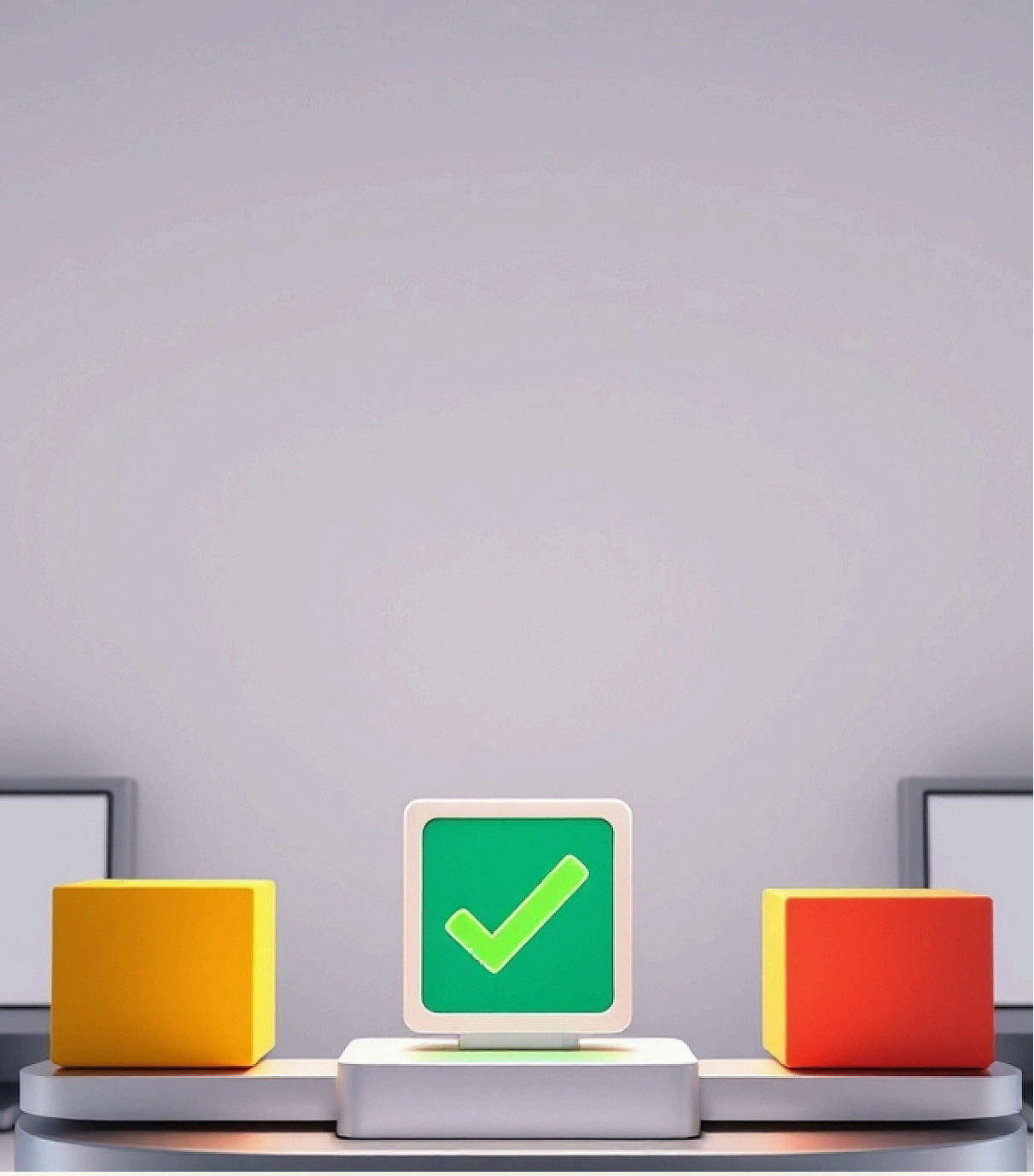
Correct the wrong step in the instructions, such as dividing by the number of students. Change the program so it divides by 3 instead of 2. The computer itself needed no repair; only the instructions were wrong. Fixing a logic error means fixing the plan, not the machine. After you change the step, the program should follow the correct plan.



Run the program again and compare its output with the expected answer. For the scores 80, 90, and 100, the program should now print 90. If it does, the logic error is gone. This confirmation step is important because it shows that your fix worked. Always check the new result; do not assume it is correct just because you changed something.



A common mistake is assuming a program without an error message must be correct. Beginners often think that if the program runs, it is fine. But a logic error can hide behind a normal-looking output. Always check the result against what you expect. Compare with known answers and try different examples. That habit helps you catch logic errors early.



A logic error means the program runs but gives a wrong result because the solution was set up wrong. The computer follows instructions faithfully, so the mistake is in the plan. To find logic errors, compare with expected results and try different examples. Check each step and fix the wrong one. Remember that a program without an error message can still be wrong.