1. **Make a comparison of static and dynamic testing techniques. Give the advantages and possible limitations when using each of them.**

| **Static Testing Technique** | **Dynamic Testing Technique** |
| --- | --- |
| Static Testing is a method performed to check defects on a code without actually executing the software application. Static testing is performed in an early stage of the software development with the purpose of avoiding errors in an early stage, since it’s easier to find the source of defects and to fix them. There are two types of techniques used:  **Review:** Performed to find the potential defects in the design of the software. It involves checking documentations, user stories, diagrams, instructions, and other documents.  **Static Analysis:** Evaluation of the code quality that is written by developers. It involves running automated testing of the code and documentation to find obvious errors and bugs. | Dynamic Testing is a type of testing which is performed to analyze the dynamic behavior of the code by running it. To perform dynamic testing the software should be written, compiled and run.  This method includes working with the software by giving input values and checking if the output is as expected, by executing test cases though manual or automated process.  So external parameters of the program can be checked: CPU load, memory usage, response time, etc., |
| **Adv 1:** Debugging is easier before moving to dynamic testing. | **Adv 1:** Covers issues related with program executions (such as runtime, memory usage, etc.) |
| **Adv 2:** Easier to identify defects that are difficult to catch during dynamic testing | **Adv 2:** Analysis features, and interactions in the various software components |
| **Adv 3:** Prevents the design or building code with inconsistencies or inaccuracies | **Adv 3:** It involves using different scenarios, providing a clear idea of how the software will behave in the “real-world” |
| **Adv 4:** Improve the quality of the development, by designing or building better code | **Adv 3:** Tests the performance of the software such as the process and response time |
| **Adv 5:** Reduces time and cost of the development and testing later on | **Adv 4:** It helps uncover a wide range of defects |
| **Adv 6:** Easier to maintaining the software code through its life because of the early detection of defects | **Lim 1:** Time-consuming and requires a high amount of resources to run the different times of tests |
| **Adv 7:** Improves communication between teams in testing and development | **Lim 2:** Defects discovered during dynamic testing require a longer time to fix due to the challenge that is it tracking back it’s origin |
| **Lim 1:** It doesn’t execute software, so it doesn’t identify any defects related to the dynamic behavior or performance of the software | **Lim 3:** Difficult to identify the cause root of a defect |
| **Lim 2:** It mainly focus on identifying errors in documentation or flaws in the logic of software, so it can miss issues related to the interactions or behavior in the software | **-** |
| **Lim 3:** Relies on human know-how and interpretation or on the work products being reviewed. Since it's subjective, different individuals could have different interpretations. | **-** |

1. **The following statement is about decision coverage:**

**When the code has a single 'IF' condition and no loops (LOOP) or switches (CASE), any test we run will result in 50% decision coverage. Which option is true about this statement?**

**a. Correct. Any test case provides 100% coverage of statements, thus covering 50% of solutions.**

**b. Correct. The result of any IF condition test will be either true or false.**

**c. Incorrect. A single test case can guarantee 25% coverage of the solutions in this case.**

**d. Incorrect, because it is too general a statement. We cannot know if it is correct as it depends on the software being tested.**

The correct answer is B. If we have more than two statements, not all test cases can cover 100% statements.

1. **There’s the following pseudocode: Switch PC on -> Start MS Word -> IF MS Word starts THEN -> Write a poem -> Close MS Word. How many test cases will it take to test its functionality?**

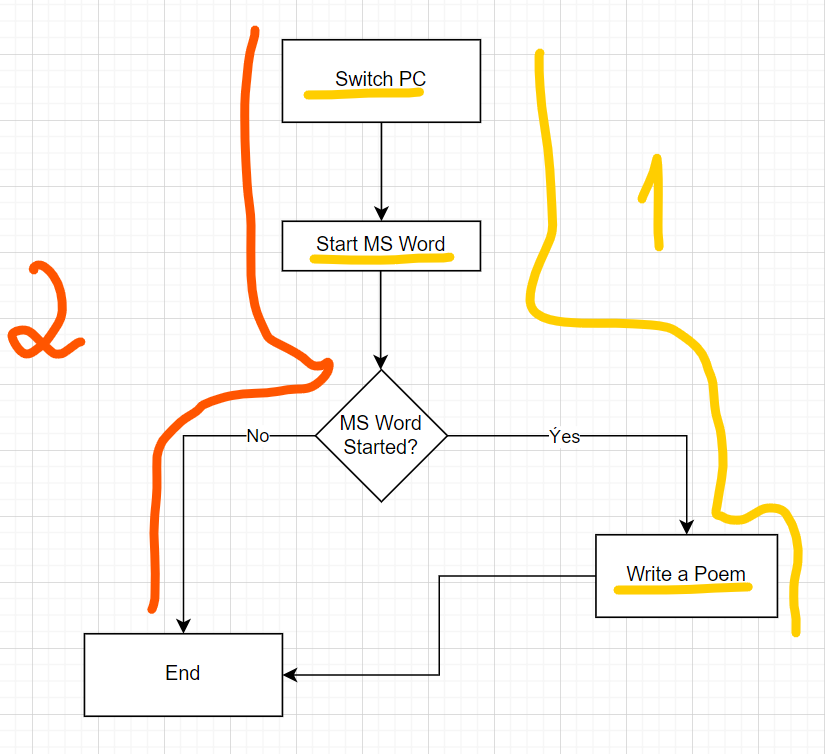
**a. 1 for operator coverage, 2 for decision coverage**

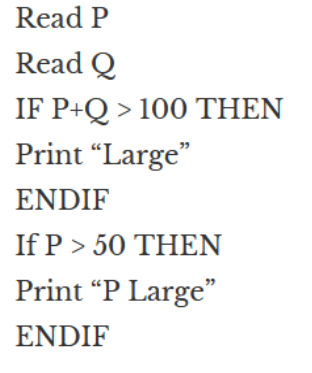
**b. 1 for operator coverage, 1 for decision coverage**

**c. 2 for operator coverage, 2 for decision coverage**

**d. 2 for operator coverage, 1 for decision coverage**

The correct answer in my opinion is A (“1 for operator coverage, 2 for decision coverage”). We just need test case 1 to run all the statements (Switch PC, Start MS Word, Write a Poem) but we need both test cases to test both conditions.



1. **How many tests are needed to check code statements:** 

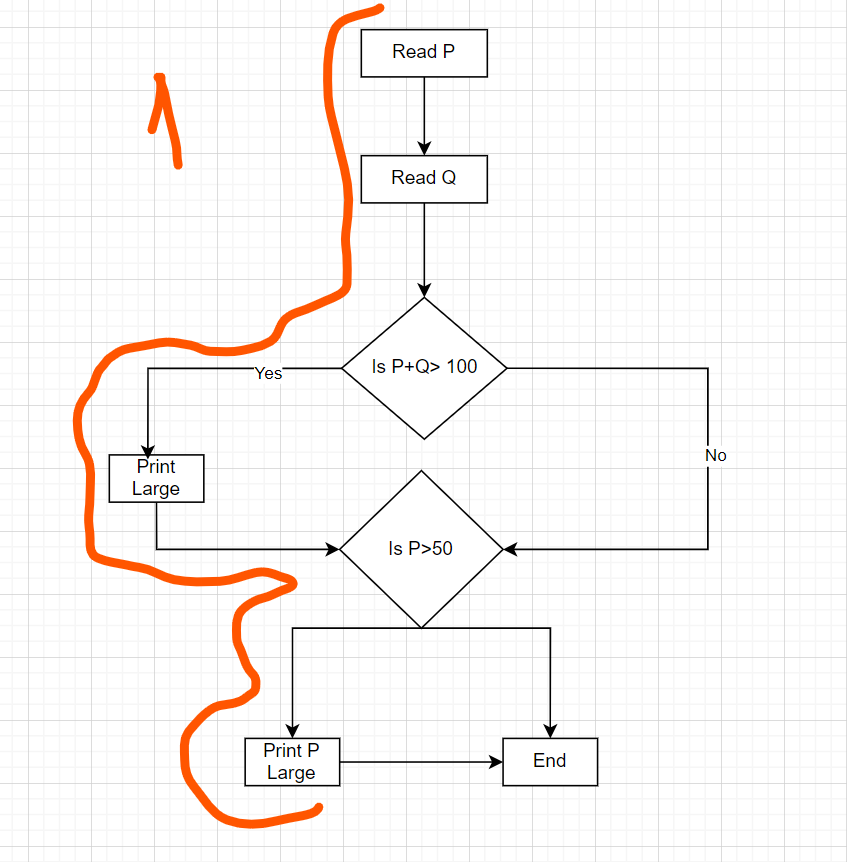
**a. 2**

**b. 1**

**c. 3**

**d. 4**

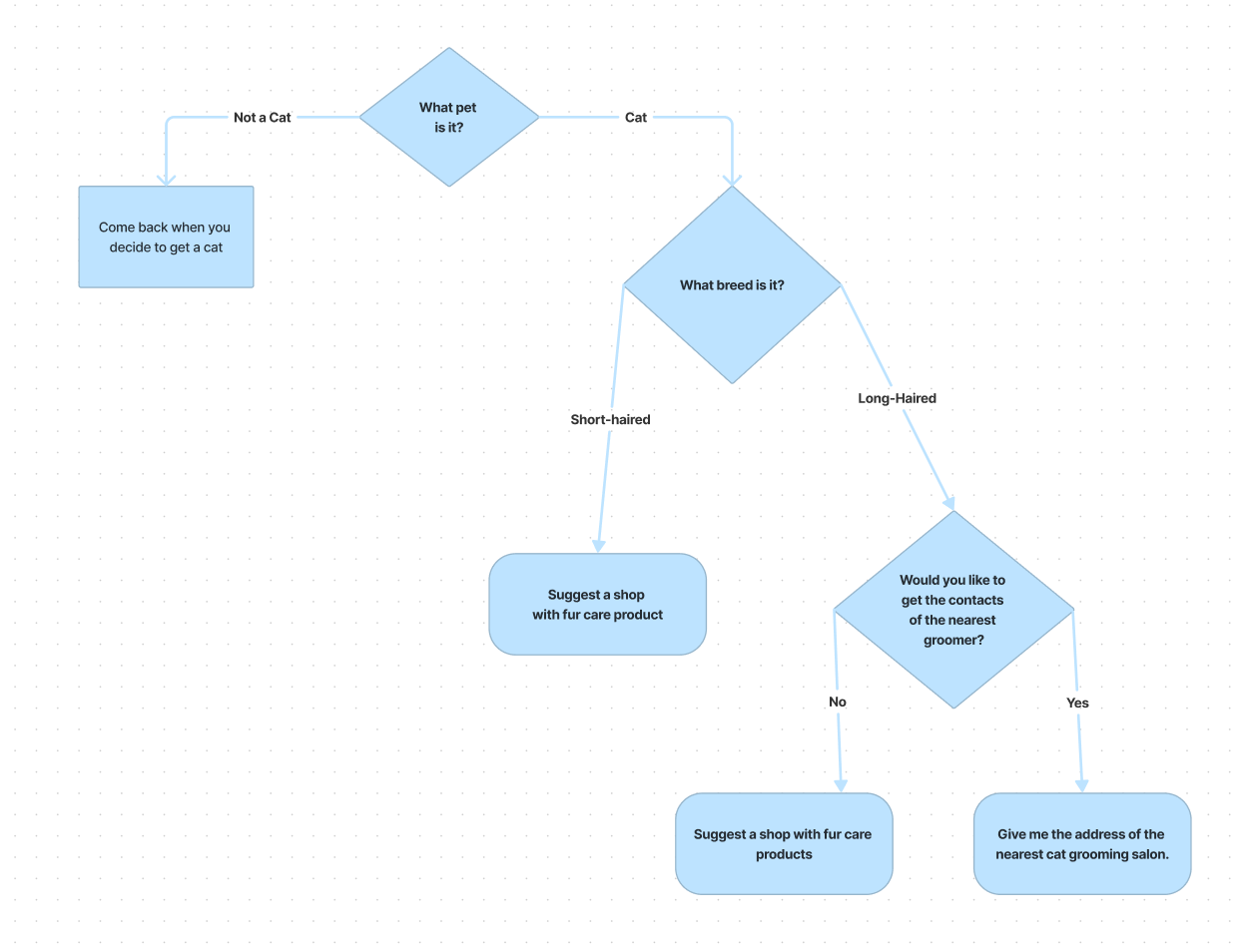
The correct answer in my opinion is b (1), according to the screenshot below. We just need one test case (in orange) to test all the statements.



1. **We continue working on a startup for a cat photo sharing app. There’s the following algorithm:**

* **Ask what kind of pet the user has.**
* **If the user answers that they have a cat, then ask what breed it is: "short-haired or long-haired?"**
* **If the user answers "long-haired", then ask: "Would you like to get the contacts of the nearest groomer?"**
* **If the user answers "yes", then say: "Give me the address of the nearest cat grooming salon."**
* **else**
* **Say: "Suggest a shop with fur care products"**
* **end**
* **else**
* **Say "Suggest a pet shop"**
* **end**
* **If the user has no cat**
* **Say "Come back when you decide to get a cat"**
* **end**
  1. **Draw an algorithm diagram (in a tool of your choice, for example, in the built-in Google Docs editor, figjam, or other.)**

<https://www.figma.com/file/SMUXGtGtinXaQ9c0FOZB71/Untitled?type=whiteboard&node-id=0%3A1&t=nbuLvh6xQ0y2GiOe-1>



* 1. **What is the minimum set of test cases needed to make sure that all questions have been asked, all combinations have been passed, and all answers have been obtained?**

We need a total of 4 test cases to test all the statements and all the decisions in the algorithm diagram.

