

97 Program Testing interview questions to hire top engineers

Questions

1. Imagine you're building a toy car. How do you make sure it moves smoothly and doesn't break easily?
2. If a program is like a recipe, how would you check if the recipe makes a tasty dish before serving it to everyone?
3. Let's say a website is like a playground. What are some things you'd check to make sure kids can play safely and have fun?
4. If you had to explain what testing is to your teddy bear, what would you say?
5. Why is testing important, even if the program seems to work perfectly at first glance?
6. What's the difference between finding a bug before someone else does, versus finding it after many users have had issues?
7. If you could only test one thing about a new app, what would it be and why?
8. Have you ever found a mistake in a game or app you use? What did you do about it?
9. What qualities would you look for in a good program tester?
10. If you see something in a program that doesn't look right, how would you describe it so someone else understands?
11. What do you understand by the terms 'Test case' and 'Test suite'?
12. Can you differentiate between functional and non-functional testing with an example?
13. Have you heard of Black box testing? Tell me what you know about it.
14. What are some of the software testing methodologies that you are aware of?
15. What is the importance of documentation in software testing, and how does it aid the testing process?
16. Explain the concept of a 'bug lifecycle' in software testing. What are the typical stages a bug goes through?
17. How do you decide which test cases to automate and which ones to execute manually?
18. What is the difference between verification and validation in software testing?
19. What is exploratory testing, and when might you use it?
20. How do you handle a situation where the developers disagree with your bug report?
21. Imagine you're testing an app that should add two numbers. What different kinds of numbers would you try?
22. What's more important: testing quickly or testing thoroughly? Why?
23. If a program works perfectly on your computer but not on someone else's, what could be the problem?
24. Can you share an example of a situation where testing saved the day?
25. Let's say you are testing a website search functionality. What potential problems or bugs would you look for?
26. What steps would you take if you found a bug that you couldn't reproduce reliably?
27. How do you prioritize which bugs to fix first when there are many to address?
28. Explain the concept of regression testing. Why is it important?
29. Have you heard of the terms 'alpha testing' and 'beta testing'? How do they differ?
30. What are your preferred testing tools, and in what scenarios are these tools useful?
31. What is a test case, and can you give a simple example?
32. Imagine you are testing a toy car. What are some things you would check to make sure it works well?
33. What does it mean to 'debug' a program, like finding a boo-boo in a game?
34. Can you describe the difference between testing something yourself and having a friend test it?
35. Why is testing important before people start using a new app or website?
36. What is a 'bug' in a program, and what happens when you find one?
37. How do you know if a test has passed or failed?
38. If a program is supposed to add two numbers, what would you test to make sure it works correctly?
39. What do you do if you find something wrong while testing?
40. Have you ever tested a game or an app before? What did you do?
41. If you could only test one thing on a new website, what would it be and why?
42. What's the difference between 'good' and 'bad' software?
43. If a program is supposed to show your name, how would you test that it is correct?
44. Let's say a website asks for your age. What ages would you try to enter to test it?
45. What is the first thing you do when you are asked to test something?
46. What does 'quality' mean to you in terms of software?
47. Explain what 'user-friendly' means when talking about a program or app.
48. If you were testing a calculator app, what buttons would you press to start?
49. What is an 'edge case', and can you give an example of one?
50. If the instructions for a program are unclear, what should you do?
51. Why do you think it's important to write down the steps you take when testing?
52. Have you ever had to follow instructions to build something? How is that similar to testing software?
53. What is the difference between expected result and actual result?
54. Can you explain the term 'test environment' in simple terms?
55. How would you decide which parts of a program to test first?
56. What is 'positive' and 'negative' testing? Give an example.
57. How would you design test cases to effectively cover various boundary conditions in a software program?
58. Can you explain the differences between white box, black box, and gray box testing techniques, and when would you choose one over the others?
59. Describe your experience with different test automation tools and frameworks. Which tools are suitable for different scenarios?
60. What are the key challenges in testing a microservices architecture, and how would you address them?
61. How do you approach testing a feature that involves complex integrations with third-party APIs?
62. Explain the concept of test-driven development (TDD) and its benefits. Have you used TDD in your projects?
63. How would you design a comprehensive test plan for a mobile application, considering different platforms and devices?
64. What are the strategies for testing software performance, including load testing, stress testing, and endurance testing?
65. Describe the process of creating and maintaining a robust test data management strategy.
66. How do you prioritize test cases when time and resources are limited? What factors do you consider?
67. Explain the concept of continuous integration and continuous delivery (CI/CD) and how testing fits into the pipeline.
68. How do you approach testing security aspects of a web application, such as authentication, authorization, and data protection?
69. What is mutation testing, and how does it help improve the quality of test suites?
70. How do you handle flaky tests in an automated testing environment? What steps do you take to identify and resolve them?
71. Describe your experience with testing different types of databases and ensuring data integrity.
72. How do you ensure that test environments are consistent and reliable across different stages of testing?
73. Explain the challenges of testing legacy systems and how you would approach modernizing the testing process.
74. How do you measure the effectiveness of your testing efforts? What metrics do you track?
75. Describe your experience with exploratory testing and how it complements other testing techniques.
76. How would you approach testing a software application for accessibility, ensuring it meets the needs of users with disabilities?
77. How do you approach testing a feature with poorly defined requirements?
78. Describe a time you had to advocate for thorough testing despite time constraints. What was the outcome?
79. How do you stay current with the latest testing methodologies and tools?
80. Explain your experience with test automation frameworks and your preferred approach.
81. What strategies do you use to identify and mitigate risks during the testing process?
82. Describe your experience with performance testing and the tools you've used.
83. How do you handle conflicting priorities when multiple projects require your testing expertise?
84. Explain your understanding of security testing principles and your approach to identifying vulnerabilities.
85. Describe a situation where you had to debug a complex issue in a production environment. What steps did you take?
86. How do you measure the effectiveness of your testing efforts?
87. Explain your experience with testing microservices architectures.
88. How do you approach testing a legacy system with limited documentation?
89. Describe your experience with testing mobile applications (iOS and Android).
90. How do you integrate testing into a CI/CD pipeline?
91. Explain your experience with testing different types of APIs (REST, GraphQL).
92. How do you handle testing data-intensive applications?
93. Describe your experience with testing cloud-based applications (AWS, Azure, GCP).
94. How do you collaborate with developers to improve code quality?
95. Explain your approach to test data management.
96. How do you determine the appropriate level of test coverage for a given project?
97. Describe a time you identified a critical bug that prevented a major release. What was the impact?