

# 109 IBM RPG interview questions to hire top developers

## Questions

---

1. What is RPG in IBM i, and why is it important for business applications?
2. Can you explain the basic structure of an RPG program?
3. What are the different data types available in RPG?
4. Explain what a 'fixed-form' RPG is.
5. What's the difference between 'free-form' RPG and 'fixed-form' RPG?
6. How do you declare a variable in RPG?
7. How do you use comments in an RPG program?
8. What are op-codes? Give some examples.
9. Explain the use of the IF and ELSE statements in RPG.
10. How do you perform calculations in RPG?
11. What is a loop in programming, and how do you implement different types of loops (e.g., FOR, DO) in RPG?
12. How do you display output to the screen in RPG?
13. What is a subfile, and why is it used?
14. Can you explain how to read data from a database file in RPG?
15. How do you update records in a database file using RPG?
16. What are the different file types you can use in an RPG program?
17. What is the purpose of the File Description Specifications (F-specs)?
18. What is the purpose of the Input Specifications (I-specs)?
19. What is the purpose of the Calculation Specifications (C-specs)?
20. What is the purpose of the Output Specifications (O-specs)?
21. What is a program status data structure (PSDS), and what information does it provide?
22. How do you handle errors in an RPG program?
23. What are indicators in RPG, and how are they used?
24. Explain the concept of modular programming in RPG.
25. What is the difference between a program and a module in RPG?
26. How do you call a sub-procedure from within an RPG program?
27. What are some of the debugging tools available for RPG programs?
28. What is SQLRPGLE in IBM i, and what are its benefits?
29. What is RPG in simple terms? Can you relate it to anything else a five-year-old might know?
30. Imagine you have a toy box. How would you tell the computer to find a specific toy using RPG?
31. What does 'compiling' mean in RPG, like when you're building with LEGOs?
32. If data is like toys, what are variables in RPG? How do you name these 'toy containers'?
33. Explain 'fixed-form' versus 'free-form' RPG like you're describing different ways to draw.
34. What is a 'subroutine' in RPG? Imagine you're teaching a robot a short dance routine.
35. What are 'indicators' in RPG? How do they help you decide what to do next in your program, like traffic lights?
36. How would you use RPG to add two numbers together? Pretend they're the number of candies you and your friend have.
37. What is a 'file' in RPG? Is it like a notebook or a computer file?
38. What is the difference between 'character', 'numeric', and 'date' data types in RPG?
39. Explain what a 'loop' is in RPG, like going around in a circle until you're dizzy.
40. What does 'debugging' mean? How do you find mistakes in your program, like finding a broken toy?
41. Can you name a few built-in functions RPG provides? What simple tasks do they help with?
42. Explain what an 'array' is. Imagine it's a shelf where you store many similar toys.
43. What is the purpose of a 'cycle' in RPG? How does RPG know when to stop?
44. What is a 'record format'? How does it describe the structure of data in a file?
45. Describe the difference between 'read' and 'write' operations in RPG.
46. What is the significance of the 'Control' specification (H-spec) in RPG?
47. What are 'named constants'? Why are they useful in an RPG program?
48. What is the purpose of the 'Move' operation (MOVE) in RPG?
49. Can you explain what a 'parameter' is in RPG? Why are parameters useful?
50. What is the 'Display File' in RPG? How is it used to show information to the user?
51. What is the 'Update' operation (UPDATE) used for in RPG?
52. How can you handle errors in RPG? What happens if something goes wrong during the program?
53. What is the difference between a 'compile-time' and 'run-time' error?
54. Describe the purpose of the 'Select' group in RPG. How does it make decisions?
55. What is the role of the 'Input' specification (I-spec) in RPG?
56. What is a 'Commitment Control'? How does it ensure data integrity?
57. Explain the difference between a subprocedure and a subroutine in RPGLE.
58. How do you handle exceptions (errors) in RPGLE programs?
59. What is the purpose of the COMMIT and ROLLBACK operations in RPGLE, and how do you use them?
60. Describe how you would use a user space in RPGLE.
61. What are the advantages of using embedded SQL in RPGLE?
62. Explain the concept of a service program in RPGLE.
63. How would you implement a recursive subprocedure in RPGLE?
64. What is the difference between a keyed file and a non-keyed file in RPGLE?
65. How can you pass parameters by reference versus by value in RPGLE?
66. Describe how you would use data structures to define complex data layouts in RPGLE.
67. What is the purpose of the OVERLAY keyword in RPGLE, and how is it used?
68. Explain how you would use the QCMDExc API to run CL commands from an RPGLE program.
69. How do you handle date and time data types effectively in RPGLE?
70. What are the benefits of using qualified data structures in RPGLE?
71. Describe a scenario where you would use the EVAL-CORR operation in RPGLE.
72. How can you dynamically allocate memory in RPGLE using the ALLOC and DEALLOC operations?
73. Explain the concept of a control break report and how you would implement it in RPGLE.
74. What is the purpose of the MONITOR group in RPGLE exception handling?
75. Describe how you would convert a DDS-defined file to a DDL-defined table and the considerations involved.
76. Explain how you can improve the performance of RPGLE programs when working with large datasets.
77. What is the difference between a shared and unshared lock in RPGLE database operations?
78. How would you use the SQL precompiler in RPGLE to prepare and execute SQL statements?
79. Describe the use of the EXFmt opcode in RPGLE.
80. Explain the different types of file feedback areas available in RPGLE.
81. How would you use the CALLB operation code in RPGLE?
82. Describe how you would implement a multi-threaded RPGLE program, and what are the potential challenges?
83. What are some strategies for debugging RPGLE programs, especially when dealing with complex logic or external calls?
84. How can you optimize RPG code for improved performance, specifically addressing issues like I/O operations and CPU usage?
85. Describe your experience with embedded SQL in RPG programs. How have you used it to interact with databases, and what are some best practices?
86. Explain the concept of service programs in RPG. How do they promote code reusability, and what are the advantages of using them?
87. What are the differences between commitment control and rollback processing in RPG, and when would you use each?
88. Discuss your experience with debugging RPG programs. What tools and techniques do you find most effective for identifying and resolving errors?
89. How have you used user interfaces with RPG, such as SDA or the web? What are the pros and cons of each approach?
90. Explain the purpose of control language (CL) programs and how they interact with RPG programs.
91. What are the different ways to handle errors and exceptions in RPG programs, and what are the best practices for error handling?
92. Describe your experience with file handling in RPG, including different file types (e.g., physical files, logical files) and access methods.
93. How do you handle date and time data in RPG programs, including formatting, calculations, and comparisons?
94. What are the advantages and disadvantages of using subfiles in RPG programs, and when would you use them?
95. Discuss your experience with data queues in RPG programs. How have you used them for inter-program communication?
96. Explain the concept of data areas in RPG and how they can be used to share data between programs.
97. How do you implement security measures in RPG programs to protect sensitive data and prevent unauthorized access?
98. What are the different ways to call RPG programs from other programming languages or systems, such as Java or web applications?
99. Describe your experience with performance tuning RPG programs. What are some common performance bottlenecks, and how do you address them?
100. How do you approach code maintenance and refactoring in RPG programs to improve readability, maintainability, and performance?
101. What are the different types of locking mechanisms available in RPG, and how do you use them to prevent data corruption in multi-user environments?
102. Explain the concept of journaling in RPG and how it can be used to recover data in the event of a system failure.
103. How do you use compiler directives and conditional compilation in RPG programs to customize code behavior based on different environments or configurations?
104. Discuss your experience with integrating RPG programs with other systems or applications, such as web services or APIs.
105. How do you use the SQL precompiler in RPG programs to optimize SQL statements and improve performance?
106. What are the different ways to handle large objects (LOBs) in RPG programs, such as images or documents?
107. Explain the concept of user-defined functions (UDFs) in RPG and how they can be used to extend the functionality of the language.