* Please give a quick introduction and overview of your experience.
* In which scenario did you use Kafka?
* Were you on the producer side of the Kafka topic?
* Where did you consume the data from Kafka?
* How do you acknowledge to Kafka that you have consumed the data?
* What is the role of the broker and the follower?
* What is the broker's purpose? Why is it needed?
* Can you create a Kafka topic without a broker?
* Do you need Zookeeper to create a Kafka topic?
* If Zookeeper was necessary, why is Kafka removing it?
* When you create a topic, where do followers come into the picture?
* What is the role of leader and follower when pushing data to a topic?
* After consuming and committing an offset, if processing fails, how do you avoid losing the data?
* If processing involves many API calls and is slow, how do you handle it without blocking consumption/commits?
* When using multithreading, how do you manage committing and processing to avoid data loss?
* Which Java version have you used?
* Are you familiar with Java Stream APIs?

For this code/quiz snippet, what would be the output? we are just assigning the same value into two different variable, and we are trying to check if two variable is equal or not using equal to equal to operator then what should be the output.

A white background with black and blue text

AI-generated content may be incorrect.

Why (explain your reasoning)?

Please think out loud and explain your thought process.

Write code to interchange key and value using Java Streams.

A white background with black text

AI-generated content may be incorrect.

* How would you handle a duplicate key collision in the map operation?
* In a key collision, should you keep the older value or replace it with the new one?
* How many types of bean injection are there in Spring?
* Can you recall any other injection types?
* Have you used transactions in Spring? Explain a scenario where you used them.
* How does Spring track and manage transactions (so it can roll back on failure)?
* Have you tried different isolation levels or propagation types in your project?
* Which propagation types have you used?
* Have you used Git?
* What is the difference between rebase and merge?
* What is squash-and-merge (git squash)?
* Which databases have you used — SQL or NoSQL?
* You mentioned limited database experience earlier — can you explain why?
* Have you used JPA repository method-name query generation?
* Do you write custom queries in JPA?
* How would you join three tables and select specific columns using JPA? (concrete example: customer, order, payment)
* How would you map repository results to a DTO in the service layer?
* Can this mapping be automated? If so, how? (hint: result set mapping)
* Have you used REST and/or SOAP web services?
* Do you know about asynchronous servlet requests?
* How would you return a non-blocking response for a GET request without blocking the servlet thread?
* What UI work have you done?
* Do you practice data structures and algorithms regularly?
* Which monitoring tools have you used (Grafana, Kibana, etc.)?
* What sort of logs do you collect?
* What metrics do you track in your monitoring tools?
* How does your team do monitoring and observability?
* How do you troubleshoot issues in production?
* Do you write scripts for operational tasks or use other tools?
* Which Spring Boot version are you using in your project?
* How would you migrate from Spring Boot 2.x to 3.x?
* What breaking changes have you encountered during migration?
* How do you handle deprecated APIs during migration?
* How do you handle global exception handling when you can’t surface runtime exceptions to the client?
* How do you normally create a new Spring Boot application? (Spring Initializer / project setup)
* What best practices do you follow when developing software or APIs?
* Have you applied SOLID principles? Can you explain them?
* Which Spring core features have you used?
* Explain the lifecycle of a Spring Bean.
* What is the difference between @Component, @Service, and @Repository?
* What is the IoC container?
* What is cyclic dependency and how can you resolve it?
* Which Spring config profiles have you used?
* Where do you store confidential configuration values?
* Do you use API mocking or virtualization?
* Which Swagger/OpenAPI version do you use?
* How can you intercept requests in a Spring Boot application?
* What are filters in an API Gateway and how do they work?
* How do you achieve API versioning?
* How do services communicate with each other in microservices (synchronous vs asynchronous)?
* Which design patterns have you used in microservices?
* Do you know about circuit breakers?
* What do open, closed and half-open states mean in a circuit breaker?
* What is the sidecar pattern?
* Have you worked with service mesh?
* Which cache server have you used?
* How do you set TTL for a specific key in a cache?
* Which caching strategies have you used (eviction policies, TTL, LRU, etc.)?
* Have you worked with graph databases?
* Is your application deployed in AWS? How do you handle load balancing?
* Which AWS services have you used?
* Have you worked with AWS Lambda?
* Do you know IAM roles?
* Have you worked with EventBridge or similar cloud integrations?
* Which databases have you worked with?
* Have you worked with JPA? How do you handle transactions in JPA?
* What are JPA transactions and how do you implement them?
* What are bidirectional relationships in JPA?
* Given an employee table (id, name, salary), write a query to get the third-highest salary.
* What are ACID properties in databases?
* What is the CAP theorem?
* What is optimistic vs pessimistic locking?
* What is a composite index and how do you create indexes?
* What is phantom read?
* What types of joins exist? What is a self-join used for?
* Have you worked with document databases?
* Have you worked with Elasticsearch or Redis?
* What does an API Gateway do?
* How do you implement authentication and authorization for a login service?
* Are you using OAuth or OAuth2?
* From a page/UI, how do you authenticate and authorize users?
* How do microservices interact with each other?
* What do you mean by API contracts / REST contracts?
* What REST design principles do you follow (verbs, idempotency, PATCH vs PUT)?
* How do you handle exceptions in microservices (ControllerAdvice / custom handlers)?
* How do you implement API security (auth server + RBAC)?
* What are the components of Kafka?
* How would you implement a Kafka client for a topic in code (template / configuration)?
* Which libraries would you add for the Kafka client?
* What considerations do you take during code review?
* What code coverage percentage do you advise maintaining?
* How would you unit-test private methods?
* How do you handle retries and republishing in Kafka?
* What is a Kafka consumer group?
* What do Kafka acknowledgments (acks 0, 1, -1) mean?
* What do you mean by idempotency and message replay?
* Which messaging queues have you used (RabbitMQ, Kafka)?
* How do you collect and query logs and where do you check logs?
* Which observability tools have you used (Dynatrace, Splunk, Grafana, Sleuth)?
* Which tools do you use in CI/CD pipelines (SonarQube, Jenkins, Bamboo, Trivy)?
* What kind of issues does SonarQube detect?
* In your current project, what role are you playing?
* Do you lead a team?
* Do you perform code reviews?
* Do you connect with stakeholders?
* Which domain have you mostly worked in?
* In technology overall, which area is your strength?