1. A corporate requirement exists that all data is to be protected on any system where it exists. What steps would you take, as a Kafka Administrator, to protect the data at rest in a Kafka Cluster you manage? (Choose Two)
    a. **You can run brokers on a machine with an encrypted file system**
    b. You can use SSL to encrypt data from the producer
    c. Kafka brokers can natively encrypt data if you set a configuration option in their server.properties file
    d. **You can programmatically encrypt data at the producer and decrypt it at the consumer**

2. Which of the following is not true about the In-Sync Replica list? (Choose One)
    a. The leader broker for a partition is always in the ISR list
    b. **Once a broker is removed from the ISR list for a partition, it cannot be added again**
    c. The ISR list contains all replicas that are identical at least up to the high-water mark
    d. When the leader broker fails, a new leader will be selected from a follower in the ISR list (unless unclean.leader.election.enabled=true)

3. Kafka Streams applications can add extra load to the Kafka cluster even though stream processing doesn't take place on the Kafka cluster itself. Which of the following increase the load on the brokers based on stream processing? (Choose Two)
    a. **Internal topics**
    b. Number of connectors
    c. **Changelog topics for local state stores**
    d. Zookeeper ephemeral nodes

4. The "magic byte" in the Kafka RecordBatch refers to which of the following? (Choose One)
    a. How many bytes are in all of the messages in the RecordBatch
    b. **The version of the message format**
    c. The sequence number of the RecordBatch from a particular producer
    d. Which compression format is used

5. When attempting to read from a subscribed topic, a consumer application is not seeing produced messages. Which of the following could be the cause? (Choose Two)
    a. **A broker on the In-Sync Replica (ISR) list has failed, delaying broker commit from a producer configured with acks=1**
    b. The consumer is unable to communicate with the __consumer_offsets topic
    c. The consumer's single call to poll() is trying to access multiple partitions
    d. **The consumer failure has caused a consumer group rebalance, which has not yet completed**

6. Several applications are producing messages with different compression formats to a single Kafka topic that has multiple partitions. How will consuming applications become aware of the different compression formats used by each producer? (Choose One)
    a. This is not possible, because each Kafka topic requires all of its partitions to have the same compression type
    b. Brokers will write each message into the partition that matches the producer's compression type setting

c. Brokers will record the compression type of each message in an internal Kafka topic
d. **Producers will indicate the compression type in the header of each message**

7. When a message is committed on a partition, the leader will advance which of the following? (Choose one)
    a. Partition log offset
    b. Log end offset
    c. **High Water Mark**
    d. Consumer Group offset

8. On rebalance, calculation of the partition assignment to the Consumer Group members is delegated to which Consumer Group entity? (Choose One)
    a. Group Partitioner
    b. Group Coordinator
    c. Group Controller
    d. **Group Leader**

9. In a Kafka cluster, when the broker running the controller thread fails, which broker will become the new controller? (Choose One)
    a. The broker having the next highest broker.id value
    b. **The next broker to successfully recreate the Zookeeper ephemeral node**
    c. The next broker to persist new metadata in Zookeeper
    d. The next broker in the cluster joined order

10. A message written to a Kafka topic results in the creation of a new online shopping order. What is the best method to update the order when a new message is received by Kafka containing a status update? (Choose One)
    a. Write a Kafka Streams application that can update/modify the order message
    b. **Write a Kafka producer that will create a new message based upon the updated order status**
    c. Write a Kafka producer that will locate and update the order message
    d. Use a Kafka connector to update the original order message