1. Why should one use Azure Key Vault when working in the Azure environment? What  
are the pros and cons? What are the alternatives?

* Pros – Increase security, apps have no direct access to keys, reduction of latency, allows for creation and import of encryption keys quickly, and allows for easy monitoring.
* Cons – Once secret keys are misplaced, the key/certificate will need to be rotated or deleted for a new instantiation.
* Alt – AWS Key Management Service (KMS), LastPass, CyberArk, KeePass

2. How do you achieve loop functionality within a Azure Data Factory pipeline? Why  
would you need to use this functionality in a data pipeline?

* ForEach functionality allows for iteration of a specific function within an API call. In this instance, we wanted to move/copy files into our Azure blob storage based on file date names. We used the iterative function to repeatedly move/store these files.

3. What are expressions in Azure Data Factory? How are they helpful when designing a  
data pipeline? Please explain with an example.

* Expressions tell Azure what types of folders/files/datatypes to look for. For example, from the image below, @{dataset().folder} indicates that the file path is a folder. The next item @{dataset()[.filename}.@{dataset().filetype}](mailto:.filename%7d.@%7bdataset().filetype%7d) indicates that we are looking for a file and a specific file type, in this case, csv.



4. What are the pros and cons of parameterizing a dataset’s activity in Azure Data  
Factory?

* You can create a single linked service (like SQL) and connect to multiple tables without creating multiple linked services for multiple tables.
* If you incorrectly parameterize the wrong item, it can cause data breaches. For example, it is not recommended to parameterize passwords/secrets. Instead those should be stored in the key vault to prevent unauthorized access.

5. What are the different supported file formats and compression codecs in Azure Data  
Factory? When will you use a Parquet file over an ORC file? Why would you choose  
an AVRO file format over a Parquet file format?

* Different supported filed formats include Avro, Binary, JSON, delimited text, Parquet, ORC, and XML.
* Parquet vs ORC – ORC stores groups of row data in stripes. Parquet stores nested data better, but ORC is better at predicate pushdown, supports acid, and more compression efficient.
* Avro vs Parquet – Avro is stored in a row format whereas Parquet is stored in a flat columnar format. Parquet will be more efficient when you need to query a few columns from the table whereas in Avro, the entire line and other unnecessary information will need to be extracted, but Avro analytical querying and write/read operations are better than Parquet.