Q1. Describe three applications for exception processing.

 **Error Handling in User Input:**

* Exception processing is used to manage invalid user inputs, such as non-numeric data when a number is expected, ensuring the program handles errors gracefully and prompts the user to correct them.

 **File Operations:**

* When working with file I/O, exception processing helps manage scenarios where files might not exist, be inaccessible, or fail to open, allowing the program to respond appropriately rather than crashing.

 **Network Communication:**

* In network applications, exception handling manages issues like connection timeouts, unreachable servers, or data transmission errors, ensuring that the application can handle network failures without crashing.

Q2. What happens if you don't do something extra to treat an exception?

 **Program Crash:**

* The application may terminate unexpectedly, leading to loss of data or unsaved work.

 **Unpredictable Behavior:**

* Unhandled exceptions can cause the application to behave unpredictably or produce incorrect results.

 **Poor User Experience:**

* Users might encounter error messages or abrupt terminations, resulting in frustration and a poor experience.

Q3. What are your options for recovering from an exception in your script?

 **Retrying the Operation:**

* Attempt to perform the operation again, which might succeed if the issue was transient.

 **Providing Fallback Behavior:**

* Implement alternative logic or default values to continue execution when the exception occurs.

 **Logging and Notification:**

* Record the error details for further investigation and notify users or administrators of the issue for appropriate action.

Q4. Describe two methods for triggering exceptions in your script.

 **Using the raise Statement:**

* Explicitly raise an exception when a specific condition occurs, e.g., raise ValueError("Invalid input").

 **Calling Functions that Raise Exceptions:**

* Invoke functions or methods that are known to raise exceptions under certain conditions, e.g., int("not a number") will raise a ValueError

Q5. Identify two methods for specifying actions to be executed at termination time, regardless of whether or not an exception exists.

 **Using finally Block:**

* Code within a finally block executes regardless of whether an exception was raised or not, ensuring cleanup actions are performed.

 **Using Context Managers (with Statement):**

* Context managers handle resource management and ensure that cleanup actions are performed when exiting the block, regardless of exceptions.