

## PYTHON

### Assignment

1. Write a program that takes input from the user, converts it to an integer, and handles `ValueError` if the input is not a valid integer.
2. Create a calculator that asks for two numbers and an operator (+, -, \*, /). Handle `ZeroDivisionError` and `ValueError` for invalid operations or inputs.
3. Write a program to open a file and handle exceptions like `FileNotFoundError` and `PermissionError`.
4. Create a custom exception class `InvalidAgeError` that is raised when the user inputs an invalid age (e.g., age below 0 or above 120).
5. Write a program that reads two integers from the user, performs division, and ensures that the result is printed regardless of whether an exception occurs or not (using `finally`).
6. Write a function that raises an exception if the input number is negative. Use `raise` to raise a `ValueError` in this case.
7. Write a program that handles both file operations and arithmetic operations within nested `try-except` blocks.
8. Write a program that takes a list and asks the user for an index, then safely retrieves the value at that index, handling the `IndexError` if the index is out of range.
9. Write a program where one exception causes another to be raised, using the `from` keyword to chain exceptions together.

10. Write a program that logs any exception raised to a text file, including the error type and message.
11. Write a function that raises a ValueError inside a nested function and observe how the exception propagates back to the calling function.
12. Create a function to calculate the square root of a number. Use an assert statement to ensure that the input is non-negative.
13. Write a custom exception class BankAccountError that stores the account number and balance when an error occurs due to insufficient balance.
14. Write a recursive function that exceeds Python's recursion limit and handle the RecursionError exception.
15. Write a custom context manager that handles specific exceptions, such as opening and closing files safely while ensuring exceptions like IOError are handled.