1. Identifying Transaction Type

Write logic:

- Read the input number.
- If the number is greater than 0, print "Positive (Deposit)".
- Else if the number is less than 0, print "Negative (Withdrawal)".
- Else, print "Zero (No Transaction)".

---

2. Summing the Digits of a Number

Write logic:

- Read the input number.
- Convert the number into individual digits.
- Initialize a sum variable to 0.
- For each digit in the number, add it to the sum variable.
- Print the sum of the digits.

3. Reversing a Transaction ID

Write logic:

- Read the input number.
- Convert the number into a string.
- Reverse the string.
- Convert it back to a number.
- Print the reversed number.

5. Finding the Factorial Using Recursion

Write logic:

- Read the input number.

- If the number is 0 or 1, return 1.
- Else, return the number multiplied by the factorial of (number - 1).

Print the result.

6. Checking if a Number is an Armstrong Number

Write logic:

- Read the input number.
- Count the number of digits.
- Initialize a sum variable to 0.
- For each digit in the number:
    - Raise the digit to the power of the total number of digits.
    - Add the result to the sum variable.
- If the sum is equal to the original number, print "Armstrong Number".
- Else, print "Not an Armstrong Number".

7. Swapping First and Last Characters of a String

Write logic:

- Read the input string.
- If the string length is less than 2, print the string as is.
- Swap the first and last characters while keeping the middle part unchanged.

Print the modified string.

8. Converting Decimal to Binary

Write logic:

- Read the input decimal number.
- Initialize an empty string for binary representation.
-

While the number is greater than 0:

    - Divide the number by 2 and store the remainder.
    - Add the remainder to the binary string.
    - Update the number by dividing it by 2.

- Reverse the binary string.
- Print the binary representation.

---

9. Finding the Longest Word in a Sentence

Write logic:

- Read the input sentence.
- Split the sentence into individual words.
- Initialize a variable to store the longest word
- Loop through each word:
- if the current word is longer than the stored longest word, update the longest word.
- Print the longest word.

---

10. Checking if Two Strings are Anagrams

Write logic:

- Read the two input strings.
- Remove spaces and convert both strings to lowercase.
- Sort the characters of both strings.
- If the sorted versions of both strings are identical, print "Anagram".

- Else, print "Not an Anagram".