1. **Scenario:** You are developing a banking application that categorizes transactions based on the amount entered.  
    Write logic to determine whether the amount is positive, negative, or zero.

**Logic:**

1. Read the given input amount
2. If the amount is greater than 0, print “Positive (Deposit)”
3. If the amount is less than 0, print “Negative (Withdrawl)”
4. Else, print “Zero (No Transaction)”
5. **Scenario:** A digital locker requires users to enter a numerical passcode. As part of a security feature, the system checks the sum of the digits of the passcode.  
    Write logic to compute the sum of the digits of a given number.

**Logic:**

1. Read the given input.
2. Initially, consider a sum variable is 0.
3. Using while loop and extracting the last digit of the number is obtained by using the modulus 10.
4. The digit is added to another variable each time the loop is executed.
5. This loop terminates when the value of the number is 0.
6. Print the total sum of the number
7. **Scenario:** A mobile payment app uses a simple checksum validation where reversing a transaction ID helps detect fraud.  
    Write logic to take a number and return its reverse.

**Logic:**

1. Read the given input
2. Convert the number into string
3. Reverse the string
4. Convert the reverse string into number
5. Print the reverse number
6. **Scenario:** In a secure login system, certain features are enabled only for users with prime-numbered user IDs.  
    Write logic to check if a given number is prime.

**Logic:**

1. Read the given number
2. Using for loop and if statement, if the given number is less than or equal to 1 and divisible by any number except 1.
3. Then print “It is not a Prime Number”
4. Else, print “It is a Prime Number”
5. **Scenario:** A scientist is working on permutations and needs to calculate the factorial of numbers frequently.  
    Write logic to find the factorial of a given number using recursion.

**Logic:**

1. Read the given input number
2. If factorial number is 0 or 1, return 1
3. Else, Using for loop and the function calls with multiply the number and number-1.
4. The function calls with decremented number until number is equal to 1 or 0.
5. Print the factorial result.
6. **Scenario:** A unique lottery system assigns ticket numbers where only Armstrong numbers win the jackpot.  
    Write logic to check whether a given number is an Armstrong number.

**Logic:**

1. Read the given input number.
2. Find the number of digits in the given number.
3. Initialize a variable sum to zero.
4. Extract each digit from the given number and raise it to the power of the number of digits and add it to the sum.
5. If the sum is equal to the given number, then print “It is an Armstrong number”
6. Else, print “It is not an Armstrong number”
7. **Scenario:** A password manager needs to strengthen weak passwords by swapping the first and last characters of user-generated passwords.  
   Write logic to perform this operation on a given string.

**Logic:**

1. Read the given input string
2. If the string has less than 2 characters then print original string itself.
3. Else, concatenated the last character, middle part, first character
4. Print the new string
5. **Scenario:** A low-level networking application requires decimal numbers to be converted into binary format before transmission.  
    Write logic to convert a given decimal number into its binary equivalent.

**Logic:**

1. Read the given decimal number
2. Initialize the empty string for binary equivalent
3. Using while loop, decimal number is divided by 2 successively and store the reminder
4. Reverse the reminder order
5. Add the reminder reverse order to binary equivalent
6. Print the binary equivalent.
7. **Scenario:** A text-processing tool helps summarize articles by identifying the most significant words.  
    Write logic to find the longest word in a sentence.

**Logic:**

1. Read the given sentence
2. Split the sentence into words using string split
3. Using for loop to iterate through the words and check the longest word by comparing their length.
4. Return the word with most characters and store as longest word
5. Print the longest word.

1. **Scenario:** A plagiarism detection tool compares words from different documents and checks if they are anagrams (same characters but different order).  
    Write logic to check whether two given strings are anagrams.  
   **Logic:**
2. Read the given two string
3. Check the length of each string are equal
4. Counts the frequency of each character in both strings.
5. If the character counts match for all characters, then print “Anagrams”
6. Else, print “Not an Anagram”