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.

**Ramishahope Artificial Intelligence Pvt Ltd**

**36, Old Anandas, SG Arcade, Marudhamalai Main Road, Vadavalli, Coimbatore -641041.**

**+91 6385383227 |** [**www.hopelearning.net**](http://www.hopelearning.net/) **|** [**mdaravind@hopelearning.net**](mailto:mdaravind@hopelearning.net) **| 33AAMCR3722R1ZU**

**Logic:**

1. Get the amount as input using ‘input’ function.
2. Use ‘if-elif’ conditions to check whether the amount is positive, negative, or zero.
3. Then print the transaction categories accordingly.
4. **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. Input the passcode.
  2. Assign 0 to variable (for eg: ‘add=0’).
  3. Use ‘for’ loop and add the digits of the number.
  4. Print the added number.

1. **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. Get the user ID using ‘input’ function.
2. Use the slicing method and get the reverse of the number as a string.
3. Convert the string back to number.
4. Print the reversed number.
5. **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. Get the user ID using ‘input’ function.
2. Use ‘if’ condition and check whether the given number is less than 2.
3. If yes, print ‘The given number is not a prime number’.
4. If it is > 2, using ‘for’ loop, check whether the number divisible in the range of 2 to square root of the number.
5. If yes, print ‘The given number is not a prime number’.
6. Else, print ‘The given number is a prime number’.
7. **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. Input the decimal number.
2. If the number is 0 or 1, return 1.
3. If the number > 1, return the number multiplied by factorial of (number -1).
4. Print the result.
5. **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. Get the lottery number using ‘input’ function.
  2. Find the length of the number.
  3. Using ‘for’ loop, find the power (equivalent to length) of each digit in the number.
  4. Add the found-out numbers.
  5. If the added number is equal to the given lottery number, then print ‘The lottery number is an Armstrong number’.

1. **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. Get the user input.
2. If the input is < 2, print the string as it is.
3. If the input is > 1, extract the first, last and middle characters.
4. Rebuild the string as: last character + middle character + first character.
5. Print the modified string.
6. **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. Input the decimal number.
  2. Create an empty string for binary representation.
  3. If the number is 0, print the binary number as 0.
  4. Use ‘while’ loop when the number > 0:

Divide the number by 2 and store the remainder.

Update the number by dividing it by 2.

* 1. Repeat the process until we reduce the number to 0.
  2. Reverse the binary string.
  3. Print the binary representation.

1. **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. Use ‘for’ loop, find the length of the words in the sentence.
2. Use another ‘for’ loop and ‘if’ condition to compare the lengths of the words.
3. Then print the longest length and the corresponding word.
4. **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:**

1. Get the strings from the user.
2. Using ‘for’ loop and ‘if’ condition, check whether the characters are the same in both the words.
3. If yes, print ‘The given strings are anagrams’.
4. Else print ‘The given strings are not anagrams’.

**Ramishahope Artificial Intelligence Pvt Ltd**

**36, Old Anandas, SG Arcade, Marudhamalai Main Road, Vadavalli, Coimbatore -641041.**

**+91 6385383227 |** [**www.hopelearning.net**](http://www.hopelearning.net/) **|** [**mdaravind@hopelearning.net**](mailto:mdaravind@hopelearning.net) **| 33AAMCR3722R1ZU**