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**

* 1. Get amount as an input
  2. If amount == 0 then Act 1, elif amount >= Act 2 else Act 3.

1. **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.
   1. Get input number.
   2. Convert input number into individual digits.
   3. Assign 0 to result variable.
   4. For each digit in the input number, add it to result variable.
   5. Print sum of digits
2. **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.
   1. Get the input number.
   2. Convert number into string.
   3. Reverse that string.
   4. Convert back to number.
   5. Print reversed number.
3. **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.
   1. Get input number.
   2. If number is less than 2, print "Not Prime".
   3. Loop from 2 to square root of that number:
   4. If number is divisible by any of these values, print "Not Prime" and exit.
   5. If not divisible by any values, print "Prime".
4. **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.
   1. Get input number.
   2. Create function using this logic. If number is 0 or 1, return 1. Else, return the number multiplied by factorial of (number - 1).
   3. 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.
   1. Get input number.
   2. Count the number of digits.
   3. Assign 0 to result variable.
   4. For each digit in the number, raise digit to the power of total number of digits and then add result to the result variable.
   5. If the sum is equal to the input number then print "It is Armstrong Number" else, print "It is not 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.
   1. Get input string.
   2. If length of input string is 1, then print the string as it is else swap the first and last characters without changing middle portion.
   3. Print the updated string.

1. **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.
   1. Get input decimal number.
   2. Declare an empty string.
   3. While decimal number is greater than 0 divide number by 2, add remainder to binary and update number by dividing it by 2.
   4. Reverse the string.
   5. Print the reversed string.

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.
   1. Get input sentence.
   2. Convert sentence to words.
   3. Assign a word to temp variable.
   4. Loop through each word. If current word is longer than temp variable word then store it to temp variable.
   5. Print temp variable string.
2. **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.
   1. Get two input strings.
   2. Remove spaces and convert both strings to lowercase or uppercase.
   3. Sort characters of both the strings.
   4. If sorted versions of both the strings are same then print "It is Anagram", else, print "It is not an Anagram".

**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**