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

* Get the amount from the user
* Check if the amount is less than 0
* If true then Print “Entered amount is negative”
* Check if the amount is greater than 0
* If true then Print “Entered amount is positive”
* Else print “Entered amount is zero”

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.

* Get the numerical code from the user
* Initialize sum variable to zero
* While numerical code is greater than 0 then calculate sum by adding it to last digit of the number using modulo operator(%)

(sum + = n%10)

* Remove the added last digit using integer division(//) from the number

n//=10

* Above process repeated until n becomes zero, so all the digit s of the number is added
* Print sum

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.  
    **Approach 1:**

* Get the number from the user
* Convert the number to string and reverse the string by slicing it in reverse order[::-]
* Convert it back to number using int()
* Print the reversed number

**Approach 2:**

* Get the number from the user
* Initialize rev variable to zero
* Check the number is greater than 0 using while loop
* If true then, retrieve the last digit from the number using modulo operator(n%10)
* Multiply rev by 10 and add it to last digit of the number and assign it to rev
* Repeat the process until n becomes zero
* Print the reversed number

1. **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

* Get the number from the user
* For i in range 2 to squareroot of number plus one
* Check If the remainder of number divided by i is equal to zero
* If true then print “given number is not prime”
* Else then print “Given number is prime”

1. **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**.**

* Get the number from the user
* Check if the number is o or 1
* If true then return 1
* Else multiple number with factorial(number-1)
* Print factorial of the given number

1. **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

* Get the number from the user
* Convert the number to string and assign it to a new variable
* Determine the length of the string using len() and assign it to new variable
* Initialize sum variable as 0
* Iterate each digit in the string and calculate the sum by adding it to int(digit)\*\*n . Here n is length of the string
* Check if the sum calculated is equal to the number
* If true then print “given number is an Armstrong number”
* Else print “Given number 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.

* Get the string from the user and assign it to variable s
* Check if length of the string is greater than 1
* If true then add last character(extract the last character using s[-1]) with middle part of the string(s[1:-1]) and first character of the string(s[0]) and assign to variable r
* Print r

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.

* Get the number from the user and assign it to num variable
* Assign bin variable to an empty string(‘ ’)
* Check if the number(num) is greater than 0
* Divide the number by 2 and convert the remainder to string
* Add the converted string with the bin variable
* Divide the number by 2 and store the quotient to num variable
* Again repeat the process until the number becomes zero
* Print bin

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.

* Get the string from the user and assign it to variable named sentence
* Split the sentence into words using split function and assign it to words variable
* Identify the longest word using max function and key parameter as len(length of the word)
* 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.

* Get the string 1 from the user and assign it to str1
* Get the string 2 from the user and assign it to str2
* Check if sorted string 1 is equal to sorted string 2 (sorted method is used to check)
* If true then print “both the strings are anagrams”
* Else print “Both the strings are not anagrams”