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**

Get the Input Transaction Amount and assign this value to the variable TransactionAmount.

Validate the TransactionAmount using If condition.

If TransactionAmount==0 then print “Amount is zero”

ElseIf TransactionAmount>0 then print “Amount is positive”

Else print “Amount is negative”.

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.

Assume passcode length should be 6 numeric values and count should be 18.

Get the numerical passcode and assign this value to the variable numPasscode.

Get the user input to enter the numerical passcode value and assign this to string variable numPasscode.

Create sum variable Intitialise as 0.

Iterating the numPasscode and get the individual value named as digit in the loop.

Convert that iterating value as Integer.

Sum = sum + int(digit)

After the loop, check the condition.

If sum == 18 then print “Passcode is match”

Else print “Passcode is not match”

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.

Get the user input to enter the numerical value and assign this to string variable inpnum.

Reverse the value using the following statement. Outnum = inpnum[::-1]

Print the outnum.

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.

Create a function named as isPrime and it has one numeric parameter as inpnum.

Inside the function.

If inpnum is less than 2 then return False

Loop the values from 2 to square root of inpnum

If inpnum modulus loopvalue==0 then return False

Return True if not fall in the above condition

Get the user input and assign to the integer variable as inpnum.

Call the above function IsPrime with the user entered argument inpnum.

Check the condition:

If the return value is true then print “given number is prime”

Else print “given number is not 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.

Create function named factorial with integer parameter n.

Inside the function:

Do the while loop n>1

Return n \* factorial(n-1)

Return 1 outside the loop

Outside the function:

Get the user int input and store that in a variable named as intinp.

Inside print function call the above function factorial with the user provided argument intinp.

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 user int input and store that in a variable named as intinp.

Get the length of intinp. leninp = len(str(intinp))

Iterate the input value:

Sum= 0

For temp as str(intinp)

Sum = sum + Int(temp)\*\*int(leninp)

If sum == intinp then print “{intinp} is an Armstrong number”

Else print “{intinp} 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 user string input and store that in a variable named as strinp.

Convert that string variable to list and store that in a list1.

By using slicing swap the first and last digits.

Assign the list1[0] first value to temp variable.

Assign the list1[-1] last value to list1[0] first value.

Assign the temp variable value to list1[-1] last value.

By using ‘’.join(list1), print the final output.

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 user input decimal named inpnum.

Create string variable named as binary for storing binary value. binary = “”

If inpnum == 0 then binary = “0”

Else

Do the while loop until inpnum > 0

Modulus the inpnum by 2 and get the reminder.

append the reminder with binary variable. Binary = str(reminder) + binary

Do the floor division. Inpnum = inpnum // 2

Print the binary variable

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 user input string named as str1 to get the sentence from the user.

By using string split function delimited by empty space, get the list with string values. Assign this to list1 variable.

By using max and len functions get the largest word.

Str2 = max(list1, key=len)

Print the largest string str2.

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.

Create a function named isAnagram and has 2 string parameters named as str1 and str2.

Inside the function.

Convert the str1, str2. Remove the blanks and convert it to lower.

Use Sorted function. It will return the list. Return sorted(str1) == sorted(str2)

Outside the function.

Get the 2 input strings and call the above function isAnagram using this 2 arguments.

If the return value is True then print “strings are Anagram”

Else print “strings are not 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**