

- **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 user input for amount.
- Use if-elif condition to check whether the amount is positive, negative, or zero.
- if the amount greater than zero, it is positive
- elif the amount is equal to zero, it is zero
- else the amount is negative.

- **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 user input for passcode and store it in a variable.
- Convert the string to a list
- Use for loop to iterate and convert the item in the list as integer and add them. Store the sum of digits in a variable.

- **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 for transaction ID.
- Use slicing [::-1] to read in reverse.
- Convert it to int() and then print the number

- **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 ID number from the user.
- Call the `is_prime` function to check whether the id is prime or not.
- `is_prime` function checks the number is not divisible by any other number other than 1 and itself. If yes, it is prime. Else, not prime. If prime, features are enabled. Else, not enabled.

- **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 to calculate factorial.
- Call factorial function and pass the number as an argument
- If the number is less than or equal to 0, it returns 1 to avoid infinite loop.
- If not multiply with that number and recursive to factorial function with minus 1 of that number.

- **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 input for a ticket number.
- Find the number of digits in that number.
- Take power to each digit of a ticket number with the number of digits in it.
- Sum all the value and store it in a variable

- Check whether the sum value is equal to ticket number. If yes, it is an armstrong number. Else, not an armstrong.
- **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 input for passwords.
 - Use swap function to swap first and last characters of that password and store it in a different variable.
- **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 for a decimal number.
 - Use bin() to convert it into binary or use loop to divide the number by 2 and then its quotient till we get 0.
- **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 for a sentence and store it in a variable.
 - Use split() to split words in a sentence.
 - Use len() to find the length of each word and use max() to compare and find the longest word.
- **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 user input for two strings and store it in two different variable.
- Call anagram check function and apply sorted() to sort each word.
- Then compare two words, if both are equal, it is an anagram. Else, not an anagram.

H O P E L A R N I N G

Ramishahope Artificial Intelligence Pvt Ltd

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

+91 6385383227 | www.hopelearning.net | mdaravind@hopelearning.net | 33AAMCR3722R1ZU