1.Scenario: You are developing a banking application that categorizes transactions based on the amount entered.

1. **Ask the user to enter a transaction amount using input()**
2. **Convert the input to an integer using int()**
3. **Store the result in a variable called transaction_amount**
4. **Check if transaction_amount is greater than 0**
5. **If true, print "Positive"; if less than 0, print "Negative"**
6. **If equal to 0, print "Zero"**

2.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.

1. **Ask the user to enter a password using input()**
2. **Store the input string in a variable called password**
3. **Initialize a variable sum_of_digits with value 0**
4. **Loop through each character in the password string**
5. **If the character is a digit, convert it to an integer and add it to sum_of_digits**
6. **After the loop, print the total sum of digits**

3.Scenario: A mobile payment app uses a simple checksum validation where reversing a transaction ID helps detect fraud.

1. **Ask the user to enter a transaction ID using input()**
2. **Convert the input to an integer using int()**
3. **Convert the integer to a string using str() and store it in num_str**
4. **Reverse the string using slicing [::-1] and store it in reversed_str**
5. **Convert the reversed string back to an integer using int()**
6. **Print the message "Reversed number:" followed by the reversed integer**

4.Scenario: In a secure login system, certain features are enabled only for users with prime-numbered user IDs.

1. **Ask the user to enter a number using input()**

2. **Convert the input to an integer using int() and store it in num**
3. **Check if num is less than 2; if true, print "Not Prime"**
4. **If num is 2 or more, start a loop from 2 to √num**
5. **Inside the loop, check if num is divisible by any i; if true, print "Not Prime" and break**
6. **If the loop completes without finding a divisor, print "Prime" using the else block of the loop**

5.Scenario: A scientist is working on permutations and needs to calculate the factorial of numbers frequently.

1. **Define a function named factorial that takes one argument n**
2. **Inside the function, check if n is 0 or 1; if true, return 1**
3. **If not, return n * factorial(n - 1) to apply recursion**
4. **Ask the user to enter a number using input() and convert it to int**
5. **Call the factorial() function with the user's input and store the result**
6. **Print the factorial result along with the original number**

6.Scenario: A unique lottery system assigns ticket numbers where only Armstrong numbers win the jackpot.

1. **Ask the user to enter a number using input() and convert it to int**
2. **Store the original number in a variable called original**
3. **Count the number of digits using len(str(num)) and store it in num_digits**
4. **Initialize a variable sum_of_powers to 0**
5. **Use a while loop to extract each digit, raise it to the power of num_digits, and add it to sum_of_powers**
6. **After the loop, compare sum_of_powers with original and print whether it's an Armstrong number**

7.Scenario: A password manager needs to strengthen weak passwords by swapping the first and last characters of user-generated passwords.

1. **Ask the user to enter a string using input()**

2. **Store the input in a variable named text**
3. **Check if the length of text is less than 2 using len(text) < 2**
4. **If true, print the original string as it is**
5. **If false, create a new string by swapping the first and last characters**
6. **Print the modified string using "Modified string:" followed by the result**

8.Scenario: A low-level networking application requires decimal numbers to be converted into binary format before transmission.

1. **Ask the user to enter a decimal number using input() and convert it to int**
2. **Initialize an empty string binary to store binary digits**
3. **Use a while loop to repeat as long as num > 0**
4. **Inside the loop, find the remainder of num % 2 and append it to binary as a string**
5. **Update num by dividing it by 2 using integer division num //= 2**
6. **After the loop, reverse the binary string and print the result**

9.Scenario: A text-processing tool helps summarize articles by identifying the most significant words.

1. **Ask the user to enter a sentence using input()**
2. **Split the sentence into words using .split() and store in words**
3. **Initialize an empty string longest to keep track of the longest word**
4. **Loop through each word in the words list**
5. **If the length of word is greater than the length of longest, update longest**
6. **After the loop, print "The longest word is:" followed by the result**

10.Scenario: A plagiarism detection tool compares words from different documents and checks if they are anagrams (same characters but different order).

1. **Ask the user to enter two strings using input()**
2. **Remove spaces from both strings using .replace(" ", "")**
3. **Convert both strings to lowercase using .lower()**

4. **Sort the characters in each string using sorted()**
5. **Compare the sorted versions of both strings**
6. **Print "Anagram" if they match, otherwise print "Not an Anagram"**