Set 2 - Python - Scenario Based - Logic

Date: 28.07.2025
Name: Prasth Varatharajan

---

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.

Answer:
1. Get amount from user
2. If the amount > 0 then print ‘positive’
3. If the amount < 0 then print ‘negative’
4. If the amount == 0 then 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.
Write logic to compute the sum of the digits of a given number.

Answer:
1. Get passcode from the user
2. If the passcode is correct then checks the sum of the passcode
3. Else print ‘incorrect password’

---

3. **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.

Answer:
1. Allow user do a transaction
2. After successful transaction get the Transaction ID
3. Create a function to reverse a string using id[::-1], then return it
4. Now store or whatever do with that reversed ID as your wish

---

4. **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

Answer:
1. Get the userID
2. Check the userID, only divided by itself and 1 with remainder 0
3. If True, Enable special features those users
4. Else, Disable those features to other users

---

5. **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

Answer:

1. Get the number from user
2. Create a recursive function to find the factorial of given number
3. Then print the output

---

6. **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.

Answer:

1. Get number from the user
2. Separate the each digits and get the count of total digits
3. Then using for loop, do (digit**count_of_digits) for each digit and add those values
4. Now check the new value is equal to the number which we get from user
5. If True, print ‘Given number is Armstrong Number’
6. Else, print ‘Given number is not Armstrong Number’
7. Ex: 153 => 1,5,3 => 1**3 + 5**3 + 3**3 => 1+125+27 => 153

---

7. **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.

Answer:

1. Get the string from user
2. Make list with password
3. Find first and last char of password
4. Swap those char using index
5. Print the new string

---

8. **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.

Answer:

1. Get number from user
2. Divide that number by 2 and we get quotient and remainder values
3. Continue the process until we get the quotient is zero
4. Now write the remainder values reverse order, it contains only 0 and 1
5. This is binary equivalent of given number and print it

---

9. **Scenario**: A text-processing tool helps summarize articles by identifying the most significant words.
Write logic to find the longest word in a sentence.

Answer:

1. Get the sentence from the user
2. Create a list with word from sentence which split by space(‘ ‘)
3. Do for loop for list and check each word length like (len(a) > len(b))
4. End of the loop we find the longest word and its length
5. Now print the word and its length

---

10. **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.

Answer:

1. Get the two words from the user
2. Check both words has same char length
3. If True, then check Word1 chars present in Word2 or not
4. If all chars passed then print ‘Its Anagram’
5. Else print ‘Its not Anagram’
6. If chars length not matched then print ‘words has different length’
