1. Scenario:

1. Get the amount for transaction from the user.
2. Get the balance amount in the account.
3. If the transaction amount is equal to balance amount then the transaction is “Zero”
4. If the transaction amount is greater than the balance amount then the transaction is “Negative”
5. If the transaction amount is smaller than the balance amount then the transaction is “Positive”

2. Scenario:

1. Get the passcode as input
2. Convert the passcode as list of digits (using list and map function)
3. Sum the list and print the sum

3. Scenario:

1. Get the transaction Id as the input
2. Convert the number into string.
3. Reverse the strings using the reverse function
4. Convert into Integer and print it

4. Scenario:

1. Get the user ID as input
2. If the user id is less than or equal to 1 then the user is not prime user
3. If the user id is divisible by the number in the range from 2 to square root of the number +1 then that number is also not the prime user id
4. Else for above both condition then the user id is prime user id.

5. Scenario:

1. Get the number as input for factorial
2. Initialize the factorial value as 1 (fact = 1)
3. For every number in the range from to 1 to number+1 multiply that number with the previous factorial value fact = i *fact (recursive)
4. Print the final fact value

6. Scenario:

1. Get the number as input
2. Get the length of the number
3. To iterate every digit convert the number to string
4. Sum the each digit of the string raised to the power of length of the number
5. If the sum is equal to the number then the number is Armstrong number
6. Else not a Armstrong number

7. Scenario:

1. Get the password as input
2. If it is string length is 2 then print as it is
3. If it is greater than 2 then swap the first and last character and print the changed password.

8. Scenario:

1. Get the number as input
2. Create empty binary string list
3. Divide the number by 2 add the reminder to the binary list
4. Update the number with the quotient number.
5. Repeat the above two steps 2 till the number is 0
6. Reverse the final binary list.
7. Print the list

9. Scenario:

1. Get the sentence as input.
2. Split the sentence as words.
3. Assign a variable to hold the longest word
4. For the each word (in the loop) find the length of the word
5. Compare the length with the previous longest word length. If it is longest than previous word length then store the longest word in the assigned variable.
6. Print the longest word.

10. Scenario:

1. Get the two strings as input.
2. Remove the spaces in the strings.
3. Sort the strings and compare if both are equal then they are Anagrams
4. If not then Not Anagrams.