Documentation of code

Task 1

Documentation for the Grocery Shopping Code:

This Typescript code provides a simple command-line interface for a grocery store shopping experience. Below is a detailed explanation of the code's flow and functionality:

Importing Dependencies:

The code begins by importing the inquirer library, which is used for taking user input in a friendly manner.

Defining Interfaces:

Two interfaces are defined: shopping and grocerychoices.

shopping is used to capture the user's selection of a shopping category, which can be either 'fruits' or 'vegetables'.

grocerychoices defines the structure for storing selected items under the 'fruits' and 'vegetables' categories.

Item Prices:

The prices variable contains the prices of various grocery items.

User Input - Shopping Category:

The user is prompted to select a shopping category ('fruits' or 'vegetables').

User Input - Grocery Items:

Depending on the selected category, the user is prompted to select specific items from a list of available options.

Calculating Total Amount:

Prices of the selected items are retrieved from the prices variable, and the total amount is calculated.

Applying Discounts:

If the total amount is equal to or greater than 300, a discount of 50 is applied.

Payment Method:

The user is prompted to select a payment method ('cash' or 'use card').

Final Output:

Based on the selected payment method, the payable amount is displayed.

This code simulates a simple grocery shopping experience, allowing the user to choose between fruits and vegetables, select specific items, calculate the total amount, apply discounts, and choose a payment method.

Task 2

Function Documentation:

Add Function: The add function takes two parameters, x and y, both of type number. It returns the sum of the two numbers.

Check Even or Odd Function:

The checkEvenOrOdd function takes a single parameter x of type number. It checks if the number is even or odd and returns a corresponding string.

Calculate Area Function:

The calculateArea function takes two parameters, height and width, both of type number. It calculates and returns the area using the formula: height * width.

Reverse String Function:

The reverseString function takes a single parameter input of type string. It reverses the characters of the input string using the split, reverse, and join methods.

Convert Celsius to Fahrenheit Function:

The convertCelsiusToFahrenheit function takes a single parameter x of type number representing the temperature in Celsius. It converts the Celsius temperature to Fahrenheit using the conversion formula and returns the result. Typescript

These functions perform specific tasks such as addition, checking if a number is even or odd, calculating the area, reversing a string, and converting temperature units

---

Task 3

Array Methods and Usage Documentation:

Push, Pop, Shift, and Unshift Methods:

An array named numbers is declared and initialized with [1, 2, 3, 4, 5, 6].

The **push method** is used to add the value 67 to the end of the array.

The **pop method** is used to remove the last element from the array.

The **shift method** is used to remove the first element from the array.

The **unshift method** is used to add the value 23 to the beginning of the array.

The resulting array is then logged to the console.

Slice Method:

An array named number is declared and initialized with [45, 34, 23, 31, 78, 65].

The slice method is used to create a new array containing elements from index 2 to 4 (excluding the element at index 5).

The resulting array is then logged to the console.

Splice Method:

An array named names is declared and initialized with ['ali', 'hassan', 'attiques', 'noman', 'Asad', 'saboor', 'taqi'].

The splice method is used to remove three elements starting from index 2 and replace them with the values 'touseeq' and 'sabtain'.

The resulting array is then logged to the console.

These examples demonstrate the usage of various array methods such as push, pop, shift, unshift, slice, and splice. Each method performs specific operations on the array, allowing for manipulation and extraction of elements based on the specified criteria.