COMPUTER PRAGRAMMING.

ASSIGNMENT # 01.

(CLO 1)

NAME: ABDUL QAYYUM (01-131232-003).

ABDUL WAHAB (01-131232-007).

SECTION : BSE 1-A.

TEACHER'S NAME: SIR RAJA M.SULEMAN.

Question 1: Finding the Shortest Path.

Imagine you are developing a GPS navigation system. You are given a map with various locations and the roads connecting them. Your task is to write an algorithm to find the shortest path from one location to another. You can assume that you have a list of locations and the distance between each pair of locations. Your algorithm should output the shortest path and the total distance.

Algorithm #1: Finding the Shortest Path.

1 **:** **START**

2 **:** Create a function called shortestPath that reads the starting location and the destination location.

3 **:** Initialize an empty list called path to store the shortest path.

4 **:** Initialize a variable called totalDistance to keep track of the total distance.

5 **:** Set the current location to the starting location.

6 **:** While the current location is not the destination location:

- Add the current location to the path.
- Find the neighboring location with the shortest distance to the destination.
- Add the distance from the current location to the neighboring location to the totalDistance.
- Set the neighboring location as the new current location.

7 **:** Add the destination location to the path.

9 **:** Return the path and the totalDistance.

10 **:** **END**

Question 2: Sorting a List of Numbers.

You are working on a project where you need to sort a list of numbers in ascending order. Design an algorithm to efficiently sort a list of integers. You should consider various sorting algorithms, evaluate their time complexity, and choose the most suitable one for the task.

Algorithm #2: Sorting a List of Numbers

1 **:** **START**

2 **:** Create a function called quickSort that takes a list as input.

3 **:** If the list has lesser than 2 elements, it is already considered sorted, so return the list.

4 **:** Select a pivot element from the list (e.g., the first, last, or middle element).

5 **:** Partition the list into two sub-arrays: elements less than the pivot and elements greater than the pivot.

6 **:** Recursively apply the quickSort function to the two sub-arrays.

7 **:** Concatenate the sorted sub-arrays and the pivot to form the final sorted list.

8 **:** Return the sorted list.

9 **:** **END**

Question 3: Calculating Fibonacci Numbers.

The Fibonacci sequence is a series of numbers where each number is the sum of the two preceding ones (e.g., 0, 1, 1, 2, 3, 5, 8, 13, ...). Write an algorithm to calculate the nth Fibonacci number. Your algorithm should be efficient and capable of handling large values of n.

Algorithm #3: Calculating Fibonacci Numbers

1 : **START**

2 : Declare a Function that reads single integer value "N" from user.

3 : Within this function initialize three long long integer variables "Previous", "Next" and "Temp".

4 : Assign the value 0 to "Previous".

5 : Assign the value 1 to "Next". Leave "Temp" unassigned.

6 : Create a For-loop with "i" assigned the value 1 which will increment until "i" is less than "N".

7 : Within this for loop:

- **a.** Assign the value in "Previous" to "Temp".
- **b.** Assign the value of "Next" to "Previous".

8 : Add the value in "Temp" with the value in "Next" and store it in "Next".

9 : The function should return "Previous".

10 : Within the main function declare a variable that will take an input from the user regarding the number of terms they want the series to go to.

11 : Finally, print the Fibonacci series using the value given by the user within the function.

12 : **END**

Question 4: Inventory Management.

You are tasked with creating an algorithm for a store's inventory management system. Your algorithm should be able to add and remove items from the inventory, update the quantity of existing items, and generate reports of the items and their quantities. Design an algorithm that efficiently manages the store's inventory based on these requirements.

Algorithm #4: Inventory Management

1 **:** **START**

2 **:** Print Menu having options:

- **a.** Add Item.
- **b.** Remove Item.
- **c.** Update Quantity.
- **d.** Generate Reports.

3 **:** Read User's choice

4 **:** If Choice is **Add Item:**

- **a.** Read item name, quantity, and any other relevant details.
- **b.** Create a new entry in the inventory with the provided information.

5 **:** If Choice is **Remove Item:**

- **a.** Read item name or unique identifier.
- **b.** Find the item in the inventory and remove it from the list

6 **:** If Choice is **Update Quantity:**

- **a.** Read item name or unique identifier and the new quantity.
- **b.** Find the item in the inventory and update its quantity with the provided value.

7 **:** If Choice is **Generate Reports:**

- **a.** Print options to generate different types of reports, such as a list of all items and their quantities, items with low stock, or items that need to be restocked soon.
- **b.** Read user's choice and print the selected report.

8 **:** **END**