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:

Step 1: start

Step 2: Read user's current location and destination point A and B.

Step 3: Go through all the locations from A to B and read each distance.

Step 4: compare all distances and save the smallest one in a variable 'shortest_distance'

Step 5: Print the name of the shortest path and its total distance that is shortest_distance

Step 6: 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:

Step 1: Start

Step 2: Read a list of integers numbers[n]

Step 3: divide the list into 2 sublists. First is the sorted list numbers[i] and second is the unsorted list numbers[j]

Step 4: Find the smallest element in the unsorted list and swap it in the sorted list.

Step 5: Repeat step 4 until you reach end of unsorted list

Step 6: 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:

Step 1: Start

Step 2: Declare an array sequence[n+1]

Step 3: Read n from user

Step 4: Initialize index 0 to 0, sequence[0]=0

Step 5: Initialize index 1 to 1, sequence[1]=1

Step 6: Initialize an iteration variable i=2 and increment i

Step 7: Perform the following operation: sequence[i]=sequence[i-1]+sequence[i-2]

Step 8: Repeat step 7 until the condition i<=n is met

Step 9: Print sequence[i]

Step 10: 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.

Algorithms:

Step 1: Start

Step 2: Create global arrays for item_name[] and item_quantity[]

Step 3: Create functions for adding and removing items, updating quantity, and generating a report. Add_item(), remove_item(), update_item_quantity(), report().

Step 4: input the following choice from user: Press ‘+’ to add item to inventory, press ‘-‘ to remove item from inventory, press ‘u’ to update quantity of existing item, press ‘p’ to print inventory report.

Step 5: If user choice==’+’:

> Input item name from user in item_name[] to add new item to inventory.

Else if user choice==’-‘:

Search for item name in the array and delete it from list of item_name[] and item_quantity[]. Then put the array back in order.

Else if user choice==’u’:

Input quantity from user and replace it in the relevant index in item_quantity[] to update.

Else if user choice==’p’:

Print the list of items and their relevant quantities to generate a report.

Step 6: End