Muhammad Omer Jawaid	01-131232-069
Muhammad Abdullah Sawar	01-131232-049
Batch Number: BSE 1A	

<u>Computer Programming</u>

<u>Assignment 1: Problem Solving (CLO-2)</u>

Question 1: Finding the Shortest Path

1. Start
2. Let A be the starting point and B be the ending point.
3. Initialize value of A to 0 and B will store the distance of each path as an array.
4. Let 3 locations loc1, loc2, loc3 be the unvisited locations.
5. For the First path A is joined to location loc1 and loc1 is to loc2 and join location loc 2 to B. This is the first possible path through the locations.
6. For the second path join A to location loc 3 and loc 3 to B.
7. For path 3rd join A to location loc 2 and loc 2 to B.
8. Initialize values which represent the distance between two points e.g., the distance from A to loc1 is stored in loc1.
9. When object travels from point A to any unvisited location the value of A is added to that unvisited location.
10. After that object goes from that unvisited location to another location the value stored in 1 unvisited location will be added to 2nd unvisited location.
11. Step 10 continues until object reaches point B and adds the value to point B.
12. Now step 8 to 10 will repeat until all path possible are covered, and B stores every value when loop of step 8,9 and 10 executes.
13. After storing all the distance values in point B, compare every value stored in B of each path with each other.
14. Find the smallest value.
15. Print the smallest value as the smallest path.
16. End

Question 2: Sorting a List of Numbers

1. Start
2. Store all the values in an array.
3. Split the array in half or nearly equal halves.
4. Repeat step 3 until each sub array contains a single value.
5. Pick two values which were separated from each other, check which value is smaller.
6. Merge the values by storing smaller value first and then the larger value.
7. Repeat step 5 and 6 for all values in sub array once.
8. Then compare the first value of the left sub array with the first value of right subarray.

9. Pick the smaller one and store in a new single array.
10. When stored then increase the counter of the array of smaller value but don't increase the counter of the array of larger value.
11. Store the smaller value in new array single array and then compare again.
12. Repeat step 10 and 11 until all values are sorted and stored in the new array.
13. Repeat step 8 to 12 until single sorted array is obtained.
14. End

Question 3: Calculating Fibonacci Numbers

1. Start
2. Declare two variables new and old to store the values of the previous two numbers.
3. Initialize new and old with 1.
4. Declare i variable as a counter and initialize it to 1.
5. Enter the placement of number and store in variable n.
6. Check if the number is 1 or 0 the print n.
7. If not, then calculate the Fibonacci number by adding new and old variables.
8. Update the value of old with new and store the added value to new variable.
9. Add 1 to i as counter.
10. Repeat step 6 ,7 and 8 until i is equal to n entered by the user.
11. Print the value of new as it holds the nth Fibonacci number.
12. End.

Question 4: Inventory Management

1. Start
2. Create a Database to store the data in it.
3. Users are given options for adding item, delete, update quantity, and generate report.
4. If user clicks add item, ask user to input item name and quantity.
5. When item name and quantity are input then store the item in database.
6. Then again give options to the user.
7. If a user wants to update the quantity, the user must input item name.
8. Once the item name is input check if it exists in inventory then give user option to update the quantity. If it does not exist return to the main menu.
9. If user wants to delete items, user have to input item name if it exists in inventory give option to delete and remove from inventory else go to main menu.
10. If a user clicks on generate report, displays a report that lists all items and their current quantities in the inventory and returns to the main menu.
11. End.