

CP ASSIGNMENT: 01.

ALGORITHMS.

Group Members:

1. Jahanzaib Aamir (01-131232-038)
2. Muhammad Hamza Imtiaz (01-131232-058)

Question 1: Finding the shortest path.

Solution:

Step 1: START.

Step 2: Assuming all the locations. Each point distance is called Edge. These Edges are defined based on their Weight (Distance).

Step 3: Find the distance between a location from each and every location using the Distance Formula = $\sqrt{(x_2 - x_1)^2 + (y_2 - y_1)^2}$. And sort them in ascending order by interchanging their values as of the shortest or largest by looping through the number of locations.

Step 4: Read\input the locations of which you have to give the shortest path.

Step 5: Now compare all the distances of the Edge which are connecting them, and search for the shortest one Edge by Comparing their Distances.

Step 6: Show output of the shortest distance.

Step 7: Also show the output of the total distance by adding all the distances using the sum formula.

Step 8: END.

Question 2: Sorting a list of numbers.

Solution:

Step 1: START.

Step 2: Taking Input by the user of numbers which we have to sort in ascending order, and store the total numbers in a constant variable of datatype int in n.

Step 3: Now apply a loop on the series which will work till order of $n(n-1)$ of integers entered for comparing, apply another loop which will iterate $(n-2)$ times, it will iterate and swap values.

Step 4: Apply a condition inside the loop and declare a variable [temp] of integer datatype. If the value entered after the first value is smaller than the first value, it will assign the smaller value to temp

variable and assign the larger value to the variable of smaller value and then interchange the values. The Loop will continue until the number to be sorted in ascending order.

Step 5: To find the time complexity for the best case of Insertion Sort is the order of N for comparing $n(n-1)$, and for swapping order of (1) .

Step 6: Show output on the screen of the sorted ascending order and Time complexity.

Step 7: END.

Question 3: Calculating Fibonacci Numbers.

Solution:

Step 1: START.

Step 2: Taking Input by the user of values in Fibonacci series and store it in a variable of datatype long long int.

Step 3: Declare two variables of datatype a, b, i, show.

a=0, b=1

Step 4: Enter the number of terms of Fibonacci's series (i).

Step 5: Print the first two terms of Fibonacci's series.

Step 6: Apply Loop:

1. Show= a + b. (a=0, b=1)
2. a = b.
3. b = Show.
4. Increase the value of "i" each time by 1.

Step 7: Print the value of Show.

Step 8: End.

Question 4: Inventory Output.

Solution:

Step 1: START.

Step 2: Taking Input by the user of number of items and their quantities.

Step 3: Show a list to user and apply a condition which allows him to select only one:

1. Add or remove items.
2. Update existing items.
3. Generate reports.
4. Exit.

Step 4: Apply a loop which iterates until the user enters 4.

Step 5:

- If user selects 1. again show him again a list and ask him to select one:
 1. Add item.
 2. Remove item.

Step 6: If user selects 1. Ask him the number and add it in the total number of items using the sum formula. Similarly, if he selects 2. Ask him the number and subtract it from the total using the difference formula.

- If the user selects 2.
Ask him which item he wants to update and then apply a condition and ask him if he want to increase the quantity or decrease if he wants to increase use the sum formula otherwise, use the difference formula and update the quantity.
- If the user selects 3.
Show him an output after updating all the items and their quantities.
- If the user selects 4.
Exit the loop.

Step 6: END.