<u>Data Structures</u>

<u>Problem-1:</u>
Enter the string to be stored and string input is stopped when newline is given immediately after string characters entered and give the corresponding character to perform a function.The string characters irrespective capital or small letters are being converted to small letter in this program.
Enter character :
 ‘S’ - (Store) - Store each alphabet into linked list return the first and last node.
 ‘P’ - (Print) - Print each alphabet with number of duplicates (traversal through linked list)
 ‘A’ - (Ascend/Sort) - Sort the alphabet based on number of duplicates (Sort)
 ‘R’ - (Remove) - Remove extra alphabet if it occurs more than T times consecutively and print the final linked list.
After entering ‘R’ you will be asked to enter the frequency a character can occur maximum times
Enter new line two times in functionality input to exit from functionality input.The letter are case insensitive when a capital letter entered it converts to a lower case
Library functions used: Unordered map to store a character and its frequency correspondingly and sort and find maximum frequency character and print it.
Memset to convert the character value to unsigned char and copies it into each of the n characters of the objects which are pointed by the string.
File uploaded as: ES18BTECH11011-1st.cpp

<u>Problem-2:</u>
Give the integer values in a single row separated by space between them and then press enter to stop taking input.Enter corresponding functionality input until, to stop taking functionality input then give two new line characters to stop giving functionality input.
Functionality:
‘M’ along with integer gives maximum and minimum of a subtree rooted at that integer entered
‘P’ along with an integer entered gives predecessor of the integer entered in the tree.
‘S’ along with an integer entered gives successor of the integer entered in the tree.
‘C’ along with two integers entered gives the least common ancestor to the two integers entered in tree. As u enter and complete one functionality u get the output correspondingly.
Library functions used: None
File uploaded as: ES18BTECH11011-2nd.cpp

<u>Problem-3:</u>
Enter the number of vertices in the graph. Then enter ‘0’ or ‘1’ to determine if the graph is directed or undirected graph.Then input the edges as follows ‘E’, source, destination and edge weight.To stop giving edges edges enter two new line characters. Then after the edges given as input enter a number between 1 to 6 to perform functionality input.
If 1 entered -returns all neighbors of v
If 2 entered -returns all vertices in Graph
If 3 entered -returns the weight of the edge between v1 and v2

If 4 entered -returns whether Graph contains a vertex
If 5 entered -returns whether the graph contains an edge between v1 and v2
Enter new line character two times to exit from the functionality input
Enter one of the values and gives the corresponding input asked and check for the output.
Library functions Used: None
File uploaded as: ES18BTECH11011-3rd.c

Problem-4:
Enter the number of vertices at start then enter the edges as follows ‘E’, source, destination, weight to stop taking input of edges enter two new lines.
Then enter the functionality input:
Enter ‘F’ to perform find the number of components and vertices in the graph created
Enter ‘S’ to find the shortest path from a source point. After entering ‘S’ enter the source point to get the distances of vertices in the graph.The number of components are printed and vertices are printed, the vertices of the graph will be printed not in increasing order as I used a DFS to traverse through the tree.When shortest path is asked to find, the connected component of the source vertices and their distances will be printed and vertices of different components in the graph are not connected, as it is directed graph from source one of the edge of the component may not be reached and infinity is printed as its distance from the source.
Library functions Used:List and list iterator and pairing the elements in the lists created and a vector to store the corresponding distance of a vertex v and update them correspondingly in the queue.
File uploaded as: ES18BTECH11011-4th.cpp
Problem-5:
Enter the number of vertices and Enter the edges as follows ‘E’ , source, destination and weight of the edge to stop giving edges press two newlines gives the output of the algorithm.If more than one component in graph exists then no spanning tree exists as all the vertices are not connected.
Library functions Used: Vector and iterator for the vector and makepair to make two integers as a pair in one vector column. Sort function to sort the edges according to weight stored in the vector.
File uploaded as: ES18BTECH11011-5th.cpp