SOHAM BALWANT
LCS2023021
COMPETITIVE CODING ASSIGNMENT -7

```
#include <bits/stdc++.h>
using namespace std;


class Solution {
public:
  void dfs(vector<string>& curr, string firstEmail,
        unordered_map<string, int>& vis,
        unordered_map<string, vector<string>>& mp) {
    vis[firstEmail] = 1;
    curr.push_back(firstEmail);

    for (auto &it : mp[firstEmail]) {
      if (!vis[it]) {
        dfs(curr, it, vis, mp);
      }
    }
  }

  vector<vector<string>> accountsMerge(vector<vector<string>>& accounts) {
    unordered_map<string, int> vis;
    unordered_map<string, vector<string>> mp;

    // Step 1: Build the Graph
    for (auto &account : accounts) {
      string firstEmail = account[1];
```

```
            for (int j = 2; j < account.size(); j++) {
                mp[firstEmail].push_back(account[j]);
                mp[account[j]].push_back(firstEmail);
            }
        }

        vector<vector<string>> ans;

        // Step 2: Perform DFS to collect connected components
        for (auto &account : accounts) {
            string name = account[0];
            string firstEmail = account[1];

            if (!vis[firstEmail]) {
                vector<string> curr;
                curr.push_back(name);
                dfs(curr, firstEmail, vis, mp);
                sort(curr.begin() + 1, curr.end());
                ans.push_back(curr);
            }
        }
        return ans;
    }
};

int main() {
    Solution s;
    int n;
```

```
cout << "Enter number of accounts: ";
cin >> n;
cin.ignore();

vector<vector<string>> accounts;

cout << "\nEnter account details:\n";
cout << "(Example: John johnsmith@mail.com john_newyork@mail.com)\n";

for (int i = 0; i < n; i++) {
   string line;
   getline(cin, line);
   stringstream ss(line);
   vector<string> account;
   string word;
   while (ss >> word) {
      account.push_back(word);
   }
   accounts.push_back(account);
}

vector<vector<string>> result = s.accountsMerge(accounts);

cout << "\nMerged Accounts:\n[\n";
for (auto &account : result) {
   cout << "  [";
   for (int i = 0; i < account.size(); i++) {
      cout << "\"" << account[i] << "\"";
```

```
            if (i != account.size() - 1) cout << ", ";
        }
        cout << "],\n";
    }
    cout << "]\n";

    return 0;
}
```

OUTPUT:

```
Merged Accounts:
[
  ["John", "john00@mail.com", "john_newyork@mail.com", "johnsmith@mail.com"],
  ["Mary", "mary@mail.com"],
  ["John", "johnnybravo@mail.com"],
]
```