**// Code dùng tham khảo (bài làm của sinh viên)**

**Cách 2:**

#include <iostream>

#include <vector>

#include <unordered\_map>

#include <string>

using namespace std;

struct NODE {

string username;

vector<string> passwords;

};

typedef unordered\_map<string, NODE> HASHTABLE;

void Insert(HASHTABLE &H, const string &user, const string &pass) {

if (H.find(user) == H.end()) {

NODE newNode;

newNode.username = user;

H[user] = newNode;

}

H[user].passwords.push\_back(pass);

}

string FindPasswords(HASHTABLE &H, const string &user) {

if (H.find(user) != H.end()) {

string result = "";

for (const string &pass : H[user].passwords) {

if (!result.empty()) {

result += " ";

}

result += pass;

}

return result;

} else {

return "Unregistered User.";

}

}

void CreateHash(HASHTABLE &H, int n) {

for (int i = 0; i < n; ++i) {

string user, pass;

cin >> user >> pass;

Insert(H, user, pass);

}

}

void OutPassword(HASHTABLE &H, int m) {

for (int i = 0; i < m; ++i) {

string user;

cin >> user;

cout << FindPasswords(H, user) << endl;

}

}

int main() {

HASHTABLE H;

int n, m;

cin >> n >> m;

CreateHash(H, n);

OutPassword(H, m);

return 0;

}