#include<iostream>

#include<bits/stdc++.h>

using namespace std;

void start();

int menu();

int k=0;

struct TrieNode

{

unordered\_map<char,TrieNode\*> child;

// 'isLast' is true if the node represents

// end of a contact

bool isLast;

TrieNode()

{

// Initialize all the Trie nodes with NULL

for (char i = 'a'; i <= 'z'; i++)

child[i] = NULL;

isLast = false;

}

};

TrieNode \*root = NULL;

void insert(string s)

{

int len = s.length();

// 'itr' is used to iterate the Trie Nodes

TrieNode \*itr = root;

for (int i = 0; i < len; i++)

{

// Check if the s[i] is already present in

// Trie

TrieNode \*nextNode = itr->child[s[i]];

if (nextNode == NULL)

{

// If not found then create a new TrieNode

nextNode = new TrieNode();

// Insert into the Map

itr->child[s[i]] = nextNode;

}

// Move the iterator('itr') ,to point to next

// Trie Node

itr = nextNode;

// If its the last character of the string 's'

// then mark 'isLast' as true

if (i == len - 1)

itr->isLast = true;

}

}

void insertIntoTrie(string contacts[],int n)

{

// Initialize root Node

root = new TrieNode();

// Insert each contact into the trie

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

insert(contacts[i]);

}

void displayContactsUtil(TrieNode \*curNode, string prefix,int len)

{

// Check if the string 'prefix' ends at this Node

// If yes then display the string found so far

if (curNode->isLast)

cout << prefix << endl;

for (char i = 'a'; i <= 'z'; i++)

{

TrieNode \*nextNode = curNode->child[i];

if (nextNode != NULL)

displayContactsUtil(nextNode, prefix + (char)i,len);

}

}

void displayContacts(string str)

{

TrieNode \*prevNode = root;

string prefix = "";

int len = str.length();

// Display the contact List for string formed

// after entering every character

int i;

for (i=0; i<len; i++)

{

// 'prefix' stores the string formed so far

prefix += (char)str[i];

// Get the last character entered

char lastChar = prefix[i];

// Find the Node corresponding to the last

// character of 'prefix' which is pointed by

// prevNode of the Trie

TrieNode \*curNode = prevNode->child[lastChar];

// If nothing found, then break the loop as

// no more prefixes are going to be present.

if (curNode == NULL)

{

cout << "No Results Found for "<< prefix<< " ";

i++;

break;

}

// If present in trie then display all

// the contacts with given prefix.

if(prefix.size()==str.size()){

cout << "Suggestions based on "<< prefix<< " are \n";

displayContactsUtil(curNode, prefix,len);

}

// cout << "Suggestions based on "<< prefix<< " are \n";

// displayContactsUtil(curNode, prefix,len);

// Change prevNode for next prefix

prevNode = curNode;

}

}

int main(){

// string name[100];

string no[100];int r;

string contacts[100];

string surname[100];

int n = sizeof(contacts)/sizeof(string);

int n1= sizeof(surname)/sizeof(string);

int n2= sizeof(no)/sizeof(string);

int check=0;

check = menu();

do{

if(check==1){

cout<<"\t\t\t\t\tFirst Name :";

cin>>contacts[k];

cout<<"\t\t\t\t\t LastName :";

cin>>surname[k];

cout<<"\t\t\t\t\t\tPhoneNO:";

cin>>no[k];

k++;

}

else if(check==2){

cout<<"if u want to search all information of all the people using attribute as name/surneme/phone no give t as 1: \n";

cout<<"\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\n";

cout<<"else if you want to implement prefix search(recommendation of all of then having common prefixes) print t as 2 \n";

int t;

cin>>t;

if(t==1) {

string temp;

// cout<<" print r= 1 to display all details in accordancce of first name and r=2 for lastname r==3 for phone no";

cout<<"\n\n\n\n\n\n\n";

cout<<"YOU WILL GET INFO OF ALL USERS HAVING THIS SAME ATTRIBUTE WHICH YOU ARE GOING TO PASS\n";

cout<<"\t\t\t\t\t\t--------------------------------------\n";

cout<<"\t\t\t\t\t\t--------------------------------------\n";

cout<<"\t\t\t\t\t\t|SEARCH COMPLETE INFO ON BASIS OF |\n";

cout<<"\t\t\t\t\t\t--------------------------------------\n";

cout<<"\t\t\t\t\t\t| [1]FIRST NAME-PRINT 1 |\n";

cout<<"\t\t\t\t\t\t| [1]LAST NAME-PRINT 2 |\n";

cout<<"\t\t\t\t\t\t| [2]PHONE NO -PRINT 3 |\n";

cout<<"\t\t\t\t\t\t--------------------------------------\n";

int r;

cin>>r;

if(r==1)

{

cout << "\t\t\t\t\t\tName :";

cin >> temp;

for (int i = 0; i < 100; i++) {

if (temp == contacts[i]) {

cout << "\t\t\t\t\t\tName is Found\n";

cout << "\t\t\t\t\t\tName : " << contacts[i]<<" "<<"Surname : "<<surname[i]<< " Phone no : " << no[i]<< endl;

}

}

}

if(r==2)

{

cout << "\t\t\t\t\t\tSurname :";

cin >> temp;

for (int i = 0; i < 100; i++) {

if (temp == surname[i]) {

cout << "\t\t\t\t\t\tName is Found\n";

cout << "\t\t\t\t\t\tName : " << contacts[i]<<" " <<"Surname : "<<surname[i]<< " Phone no : " << no[i]<< endl;

}

}

}

if(r==3)

{

cout << "\t\t\t\t\t\tPHONE NO :";

cin >> temp;

for (int i = 0; i < 100; i++) {

if (temp == no[i]) {

cout << "\t\t\t\t\t\tName is Found\n";

cout << "\t\t\t\t\t\tName : " << contacts[i]<<" " <<"Surname : "<<surname[i]<< " Phone no : " << no[i]<< endl;

}

}

}

}

else if(t==2){

cout<<"PRFIX SEARCH\n";

cout<<"\t\t\t\t\t\t--------------------------------------\n";

cout<<"\t\t\t\t\t\t--------------------------------------\n";

cout<<"\t\t\t\t\t\t|PREFIX COMPLETE INFO ON BASIS OF |\n";

cout<<"\t\t\t\t\t\t--------------------------------------\n";

cout<<"\t\t\t\t\t\t| [1]FIRST NAME-PRINT 2 |\n";

cout<<"\t\t\t\t\t\t| [1]LAST NAME-PRINT 3 |\n";

cout<<"\t\t\t\t\t\t| [2]PHONE NO -PRINT 4 |\n";

cout<<"\t\t\t\t\t\t--------------------------------------\n";

cin>>r;

if(r==2){

insertIntoTrie(contacts, n);

string query;

cin>>query;

displayContacts(query);

}

else if(r==3){

insertIntoTrie(surname, n1);

string q1;

cin>>q1;

displayContacts(q1);

}

else if(r==4){

insertIntoTrie(no, n2);

string q2;

cin>>q2;

displayContacts(q2);

}

}

}

check=menu();

}while(check!=3);

}

int menu(){

cout<<"\n\n\n\n\n\n\n";

cout<<"\t\t\t\t\t\t--------------------------------------\n";

cout<<"\t\t\t\t\t\t--------------------------------------\n";

cout<<"\t\t\t\t\t\t| PHONE BOOK APPLICATION |\n";

cout<<"\t\t\t\t\t\t--------------------------------------\n";

cout<<"\t\t\t\t\t\t| |\n";

cout<<"\t\t\t\t\t\t| [1]ADD CONTACTS |\n";

cout<<"\t\t\t\t\t\t| [2]SEARCH CONTACTS |\n";

cout<<"\t\t\t\t\t\t--------------------------------------\n";

int a;

cin>>a;

return a;

}