Rijul has N contacts saved on his mobile number. However, to search a single name he doesn’t applies linear search neither uses the contacts app and types the name rather the dialer has some interesting feature. Whenever Rijul presses any number key(s), (say 12) all the names related to key 1 and 2 appear on screen. The keypad is designed in such a way that every number key has some alphabets related to it, and whenever some number keys are pressed all the names starting with the all possible combinations of alphabets formed by the pressed numbers keys appear on the screen.

The number keys have the following alphabets associated with them:

1: a, j, s

2: b, k, t

3: c, l, u

4: d, m, v

5: e, n, w

6: f, o, x

7: g, p, y

8: h, q, z

9: i, r

You have to find out all the possible names that are associated with the number keys pressed.

**INPUT FORMAT:**

The first line contains an integer N which is the number of contacts saved on Rijul’s mobile.

Following N lines contains the names of contacts.

The last line contains a string of numbers which shows the keys pressed (in order).

**OUTPUT FORMAT:**

The list of all names in a space separated manner.

**Constraints :**

len(str) < 10  
No of strings in the vector < 10

**SAMPLE INPUT**

3

clutch

rudra

salmon

34

**SAMPLE OUTPUT**

rudra

salmon

**EXPLANATION**

**34** will result into combinations :

cd cm cv

ld lm lv

ud um uv

Corresponding strings are output.

1. salmon contains **lm**
2. rudra contains **ud**

**SOLUTION**

#include <bits/stdc++.h>

using namespace std;

char table[][10] = { " ", "ajs", "bkt", "clu", "dmv", "enw" , "fox", "gpy" , "hqz", "ir" };

void give(char out[10], string str)

{

    int e=0;

    for(int s=0; str[s]!='\0'; s++)

    {

        if(out[e] == '\0')

{

cout<<str<<endl;

return;

}

        if(e>0 && out[e]!=str[s]) e = 0;

        else if(out[e] == str[s]) e++;

    }

if(out[e] == '\0') cout<<str<<endl;

}

void compute(char inp[10], char out[10], int i, int j, vector<string> v)

{

    if(inp[i] == '\0')

    {

        out[j] = '\0';

        //cout << out << " ";

        for(auto x : v) give(out, x);

        return;

    }

    int digit = inp[i] - '0';

    for(int k=0; table[digit][k]!='\0'; k++)

    {

        out[j] = table[digit][k];

        compute(inp, out, i+1, j+1, v);

    }

}

signed main()

{

    int n;

    cin >> n;

    vector<string> v(n);

    for(int i=0; i<n; i++)

    {

        string s; cin >> s;

        v[i] = s;

    }

    char inp[10];

    cin>>inp;

    char out[10];

    compute(inp, out, 0, 0, v);

}

**TESTCASE 1:**

**INPUT**

9

arjun

jasraj

aastha

parv

nishta

jaskirat

sasly

ssjain

utsav

111

**OUTPUT**

aastha

jasraj

jaskirat

sasly

ssjain

**TESTCASE 2:**

**INPUT**

10

prateek

deepak

mhir

shikha

vidhik

palak

utkarsh

divyam

sparsh

akku

489

**OUPUT**

vidhik

mhir