

Όνομα Άσκησης: Κινητό Τηλέφωνο (Name That Number)

Πηγή: Usaco

Εκφώνηση

Γνωρίζουμε ότι πατώντας τα πλήκτρα του κινητού μπορούμε να σχηματίσουμε λέξεις. Αναλόγως με το πλήκτρο που πατούμε μπορούμε να προσθέσουμε τα ανάλογα γράμματα όπως φαίνονται πιο κάτω (εκτός από τα 'Q' και 'Z':

2: A, B, C	5: J, K, L	8: T, U, V
3: D, E, F	6: M, N, O	9: W, X, Y
4: G, H, I	7: P, R, S	

Πατώντας τα πλήκτρα 4734 μπορούμε να δημιουργήσουμε όλα τα πιο κάτω ονόματα:

```
GPDG GPDH GPDI GPEG GPEH GPEI GPDF GPFH GPGI GRDG GRDH GRDI
GREG GREH GREI GRFG GRFH GRFI GSDG GSDH GSDI GSEG GSEH GSEI
GSFG GSFH GSFI HPDG HPDH HPDI HPEG HPEH HPEI HPFG HPFH HPFI
HRDG HRDH HRDI HREG HREH HREI HRFH HRFI HSDG HSDH HSDI
HSEG HSEH HSEI HSFG HSFH HSFI IPDG IPDH IPDI IPEG IPEH IPEI
IPFG IPFH IPFI IRDG IRDH IRDI IREG IREH IREI IRFG IRFH IRFI
ISDG ISDH ISDI ISEG ISEH ISEI ISFG ISFH ISFI
```

Από όλα τα πιο πάνω, το μόνο όνομα που είναι αποδεκτό είναι το GREG. Για να προσδιορίσετε τα αποδεκτά ονόματα σας παρέχεται το αρχείο dict.txt το οποίο περιέχει μια λίστα με λιγότερα από 5,000 αποδεκτά ονόματα ταξινομημένα αλφαβητικά.

Να γράψετε ένα πρόγραμμα το οποίο να δέχεται τα πλήκτρα που έχουν πατηθεί (μέχρι 12 πλήκτρα) και ακολούθως να επιστρέφει όλα τα αποδεκτά ονόματα που μπορούν να δημιουργηθούν ή 'NONE' αν δεν δημιουργείται κανένα.

Δεδομένα εισόδου (namenum.in)

Μια γραμμή που αποτελείται από τους αριθμούς 2 μέχρι 9, από ένα μέχρι δώδεκα ψηφία

Δεδομένα εξόδου (namenum.out)

Μια λίστα με ονόματα που μπορούν να δημιουργηθούν, ένα σε κάθε γραμμή, ταξινομημένα σε αλφαβητική σειρά

Παράδειγμα εισόδου

62742

Παράδειγμα εξόδου

MARIA
NASHA
NASIA

Επεξήγηση

Αφού έχουμε 3 γράμματα για κάθε ψηφίο και μέχρι 12 ψηφία σημαίνει έχουμε το πολύ $3^{12} = 531441$ λέξεις που μπορούν να δημιουργηθούν. Ακολουθώς πρέπει να ψάξουμε αν αυτές οι λέξεις βρίσκονται στο αρχείο dict.txt το οποίο μπορεί να αποβεί κάπως χρονοβόρο.

Μια δεύτερη προσέγγιση είναι να βρούμε τις λέξεις του αρχείου dict.txt που ταυτίζονται με το συνδυασμό πλήκτρων που έχει δοθεί. Ελέγχουμε αν ο αριθμός των πλήκτρων που έχουν πατηθεί είναι ίσος με το μέγεθος της λέξης και έτσι περιορίζουμε την αναζήτηση αφού δεν χρειάζεται να ελέγξουμε μικρότερες ή μεγαλύτερες λέξεις.

Λύση #1

```
#include <fstream>
#include <string>
#include <vector>
#include <algorithm>
#include <set>
using namespace std;

bool check(const string &s1, const string &s2) {
    return s1.size() == s2.size() ? s1 < s2 : s1.size() < s2.size();
}

int main() {
    bool good, match;
    string word;
    vector<string> dictionary;
    set< pair<char, char> > s;

    ifstream dict("dict.txt");
    ifstream fin("namenum.in");
    ofstream fout("namenum.out");

    while (dict >> word) {
        dictionary.push_back(word);
    }

    sort(dictionary.begin(), dictionary.end(), check);
```

```

s.insert(make_pair('A', '2'));
s.insert(make_pair('B', '2'));
s.insert(make_pair('C', '2'));
s.insert(make_pair('D', '3'));
s.insert(make_pair('E', '3'));
s.insert(make_pair('F', '3'));
s.insert(make_pair('G', '4'));
s.insert(make_pair('H', '4'));
s.insert(make_pair('I', '4'));
s.insert(make_pair('J', '5'));
s.insert(make_pair('K', '5'));
s.insert(make_pair('L', '5'));
s.insert(make_pair('M', '6'));
s.insert(make_pair('N', '6'));
s.insert(make_pair('O', '6'));
s.insert(make_pair('P', '7'));
s.insert(make_pair('R', '7'));
s.insert(make_pair('S', '7'));
s.insert(make_pair('T', '8'));
s.insert(make_pair('U', '8'));
s.insert(make_pair('V', '8'));
s.insert(make_pair('W', '9'));
s.insert(make_pair('X', '9'));
s.insert(make_pair('Y', '9'));

fin >> word;

match = false;

for (int i = 0, sz = dictionary.size(), len = word.size(); i < sz; i++) {
    if (dictionary[i].size() < len)
        continue;

    if (dictionary[i].size() > len)
        break;

    good = true;
    for (int j = 0; j < len; j++) {
        if (s.find(make_pair(dictionary[i][j], word[j])) == s.end()) {
            good = false;
            break;
        }
    }

    if (good) {
        fout << dictionary[i] << endl;
        match = true;
    }
}

if (!match)
    fout << "NONE" << endl;

return 0;
}

```

Λύση #2

```
#include <fstream>
#include <string>

using namespace std;

int press[] = {2,2,2,3,3,3,4,4,4,5,5,5,6,6,6,7,7,7,8,8,8,9,9,9};

string name_num(string num){
    string buffer = "";
    for(int i = 0; i < num.size(); ++i){

        if(num[i] == 'Q' || num[i] == 'Z')
            continue;

        if(num[i] < 'Q')
            buffer += press[int(num[i] - 'A')] + '0';
        else
            buffer += press[int(num[i] - 'B')] + '0';
    }
    return buffer;
}

int main() {
    ifstream fin ("namenum.in"), fin2 ("dict.txt");
    ofstream fout ("namenum.out");

    string number;
    fin >> number;

    bool found = false;

    string word;

    while(fin2 >> word)
        if(word.size() == number.size() && name_num(word) == number){
            found = true;
            fout << word << endl;
        }

    if(!found)
        fout << "NONE" << endl;

    return 0;
}
```