

```
#include <iostream>
using namespace std;
#include <fstream>
#include <string>
#include <algorithm>
#define SP ' '

struct Codifica{
    string CODIF;
    int    posCOD;    //-- posizione originaria nel file = valore decodifica
    int    lenCOD;    //-- lunghezza della codifica per posCOD
};

fstream fr;
fstream fw;

string CONTATTO;
int    lenCONT;
char NUMDECOD[1000];

Codifica CODICE [10];
char  num      [10] = {'0', '1', '2', '3', '4', '5', '6', '7', '8', '9'};

int N, cifreC, posC, k, lenMIN, lenMAX;

char found    = 'N';
char diverso  = 'N';

bool confronta(const Codifica& uno, const Codifica& due){

    if (uno.lenCOD > due.lenCOD)
        return true;
    else
        if (uno.lenCOD == due.lenCOD
            && (uno.CODIF  > due.CODIF))
            return true;
        else
            return false;
}
```

```
int main(){

    fr.open("input.txt", ios::in);
    fw.open("output.txt", ios::out);

    fr >> N;          //--- 1^ lettura per arrivare ai 10 codici
    for (int i=0; i<N; i++)
        fr >> CONTATTO;    //--- 1^ lettura per arrivare ai 10 codici

    lenMIN = 1001;
    lenMAX = 0;

    for (int i=0; i<10; i++) {

        fr >> CODICE[i].CODIF ;

        CODICE[i].posCOD = i;
        CODICE[i].lenCOD = CODICE[i].CODIF.length();

        if (CODICE[i].lenCOD < lenMIN)
            lenMIN = CODICE[i].lenCOD;
        if (CODICE[i].lenCOD > lenMAX)
            lenMAX = CODICE[i].lenCOD;
    }

    sort (CODICE, CODICE+10, confronta);

    fr.close();
    fr.open("input.txt", ios::in);

    fr >> N;          //--- 2^ lettura per elaborare 1 contatto alla volta

    for (int i=0; i < N; i++) {          //-1--- per ognuno degli N contatti

        fr >> CONTATTO;
        lenCONT = CONTATTO.length();

        NUMDECOD[0] = '\0';
```

```
cifreC = 0; //-- numero di cifre decodificato
posC=0;     //-- posizione da cui partire su CONTATTO per estrarre

while (posC < lenCONT) {                               //-2--- while su posC

    found = 'N';
    for (k=0; k<10 && found == 'N'; k++) {              //-3--- ciclo sui 10 codici

        if (posC + CODICE[k].lenCOD <= lenCONT) {      //-4--- lungth codice contenuta nel contatto

            diverso = 'N';
            for (int t=0; t < CODICE[k].lenCOD && diverso == 'N'; t++) { //-5--- analisi carattere per carattere

                if (CONTATTO[t+posC] != CODICE[k].CODIF[t])
                    diverso = 'S';
            }                                           //-5---END---

            if (diverso == 'N') {                       //-6--- codice contenuto nel contatto
                NUMDECOD [cifreC] = num[CODICE[k].posCOD];
                cifreC++;
                posC += CODICE[k].lenCOD;
                found = 'S';
            }                                           //-6---END---
        }
    }
}

NUMDECOD[cifreC] = '\0';
fw << NUMDECOD << endl;                               //-2---END--- analisi contatto i-esimo completata

}                                                       //-1---END--- completata analisi N contatti

fr.close();
fw.close();
}
```