

```

///Vasiluta Alexandru
#include <bits/stdc++.h>
using namespace std;
map<string,int> elem = {"A", 0xCCCC}, {"~A", 0x3333},
    {"B", 0xFF00}, {"~B", 0x00FF},
    {"C", 0x6666}, {"~C", 0x9999},
    {"D", 0x0FF0}, {"~D", 0xF00F});
string s, letters = "0123456789ABCDEF";

int getNibble(int n, int nn)
{
    int nibble = n >> ((nn - 1) * 4);
    return nibble & 15;
}

inline bool exists(set<string> s, string a)
{
    return (s.find(a) != s.end());
}

int main()
{
    while(cin >> s)
    {
        int merged = 0;
        string current;
        s += "+";
        set<string> tElem;
        for(const auto &c : s)
        {
            if(c == '+')
            {
                for(char i = 'A'; i <= 'D'; i++)
                    if( exists(tElem, string() + i) && exists(tElem, string("~") + i))
                        { tElem.erase(string() + i);

```

```
        tElem.erase(string("~") + i);}
int table = 0xFFFF;
for(auto &i : tElem)
    table &= elem[i];
if(tElem.empty())
    table = 0;
tElem.clear();
merged |= table;
continue;
}
current += c;
if(c == '~')
    continue;
tElem.insert(current);
current.clear();
}
cout << letters[getNibble(merged, 4)]
    << letters[getNibble(merged, 3)]
    << letters[getNibble(merged, 2)]
    << letters[getNibble(merged, 1)]
    << "\n";
}
return 0;
}
```