```
///Popa Sebastian
#include <bits/stdc++.h>
using namespace std;
map<int, vector<pair<int, int>>> gen_poz_map();
map<int, vector<pair<int, int>>> pozs = gen_poz_map();
map<char, int> gen_id_map();
map<char, int> idma = gen_id_map();
string solve();
vector<pair<int, int>> intersection(map<int, bool>);
char tohex(int);
int main()
{
   ios_base::sync_with_stdio(false);
   cin.tie(nullptr);
   const int test_cases = 5;
   for (int Case = 1; Case <= test_cases; Case++)
      cout << solve() << '\n';
   return 0;
}
string solve()
{
   string str, af;
   getline(cin, str);
   vector<vector<bool>> mymat(5, vector<bool>(5));
   map<int, bool> act;
   for (int i = 0, plus = 1; i < str.size(); i++)
      if (str[i] == '+')
      {
         for (int tt = 1, mm; tt <= 4; tt++)
            mm = min(act[tt], act[-tt]), act[tt] -= mm, act[-tt] -= mm;
         for (auto j : intersection(act))
            mymat[j.first][j.second] = 1;
         act.clear();
      }
```

```
        else if (str[i] == '~')
            plus = -1;
        else if (isalpha(str[i]))
            act[idma[str[i]] * plus] = 1, plus = 1;
    for (int tt = 1, mm; tt <= 4; tt++)
        mm = min(act[tt], act[-tt]), act[tt] -= mm, act[-tt] -= mm;
    for (auto j : intersection(act))
        mymat[j.first][j.second] = 1;
    for (int i = 1, num = 0; i <= 4; af += tohex(num), num = 0, i++)
        for (int j = 4, p = 1; j >= 1; j--, p *= 2)
            num += p * mymat[i][j];
    return af;
}
vector<pair<int, int>> intersection(map<int, bool> ma)
{
    vector<pair<int, int>> aux(100);
    vector<int> vx;
    for (auto i : ma)
        if (i.second)
            vx.push_back(i.first);
    if (vx.size() == 0)
        return vector<pair<int, int>>();
    if (vx.size() == 1)
    {
        aux.clear();
        for (auto i : pozs[vx[0]])
            aux.push_back(i);
        return aux;
    }
    vector<pair<int, int>>::iterator it = set_intersection(pozs[vx[0]].begin(), pozs[vx[0]].end(),
pozs[vx[1]].begin(), pozs[vx[1]].end(), aux.begin());
    aux.resize(it - aux.begin());
    for (int i = 2; i < vx.size(); i++)
    {
```

```
        vector<pair<int, int>> vir(30);
        it = set_intersection(pozs[vx[i]].begin(), pozs[vx[i]].end(), aux.begin(), aux.end(), vir.begin());
        vir.resize(it - vir.begin());
        aux = vir;
    }
    return aux;
}
map<int, vector<pair<int, int>>> gen_poz_map()
{
    map<int, vector<pair<int, int>>> ma;
    ma[1].push_back({1, 1}), ma[1].push_back({1, 2}), ma[1].push_back({2, 1}), ma[1].push_back({2,2});
    ma[1].push_back({3, 1}), ma[1].push_back({3, 2}), ma[1].push_back({4, 1}), ma[1].push_back({4,2});
    ma[-1].push_back({1, 3}), ma[-1].push_back({1, 4}), ma[-1].push_back({2, 3}), ma[-].push_back({2, 4});
    ma[-1].push_back({3, 3}), ma[-1].push_back({3, 4}), ma[-1].push_back({4, 3}), ma[-1].push_back({4, 4});
    ma[2].push_back({1, 1}), ma[2].push_back({1, 2}), ma[2].push_back({1, 3}), ma[2].push_back({1, 4});
    ma[2].push_back({2, 1}), ma[2].push_back({2, 2}), ma[2].push_back({2, 3}), ma[2].push_back({2, 4});
    ma[-2].push_back({3, 1}), ma[-2].push_back({3, 2}), ma[-2].push_back({3, 3}), ma[-2].push_back({3, 4});
    ma[-2].push_back({4, 1}), ma[-2].push_back({4, 2}), ma[-2].push_back({4, 3}), ma[-2].push_back({4, 4});
    ma[3].push_back({1, 2}), ma[3].push_back({1, 3}), ma[3].push_back({2, 2}), ma[3].push_back({2, 3});
    ma[3].push_back({3, 2}), ma[3].push_back({3, 3}), ma[3].push_back({4, 2}), ma[3].push_back({4, 3});
    ma[-3].push_back({1, 1}), ma[-3].push_back({1, 4}), ma[-3].push_back({2, 1}), ma[-3].push_back({2, 4});
    ma[-3].push_back({3, 1}), ma[-3].push_back({3, 4}), ma[-3].push_back({4, 1}), ma[-3].push_back({4, 4});
    ma[4].push_back({2, 1}), ma[4].push_back({2, 2}), ma[4].push_back({2, 3}), ma[4].push_back({2, 4});
    ma[4].push_back({3, 1}), ma[4].push_back({3, 2}), ma[4].push_back({3, 3}), ma[4].push_back({3, 4});
    ma[-4].push_back({1, 1}), ma[-4].push_back({1, 2}), ma[-4].push_back({1, 3}), ma[-4].push_back({1, 4});
    ma[-4].push_back({4, 1}), ma[-4].push_back({4, 2}), ma[-4].push_back({4, 3}), ma[-4].push_back({4, 4});
    for (int i = -4; i <= 4; i++)
        sort(ma[i].begin(), ma[i].end());
    return ma;
}
map<char, int> gen_id_map()
{
    map<char, int> ma;
    int nrx = 0;
```

```
    for (char t = 'A'; t <= 'D'; t++)
        ma[t] = ++nrx;
    return ma;
}
char tohex(int nr)
{
    if (nr <= 9)
        return '0' + nr;
    return 'A' + nr - 10;
}
```