

```

///Criveanu Bogdan
#include <bits/stdc++.h>
#define pp pair<string,vector<vector<bool>>>
#define str_mat string,vector<vector<bool>>
#define parc map<str_mat>:: iterator
using namespace std;

ifstream in;
ofstream out;

map< string,vector<vector<bool>> > lit;
string cif="0123456789ABCDEF";

void mtr1(vector<vector<bool>> &tab,vector<bool> line)
{
    for(int i=0;i<4;i++)
        tab.push_back(line);
}

void B_form(vector<vector<bool>> &tab,vector<bool> &line)
{
    tab.clear();
    line.clear();
    line={1,1,1,1};
    tab.push_back(line);
    tab.push_back(line);

    line.clear();
    line={0,0,0,0};
    tab.push_back(line);
    tab.push_back(line);

    lit.insert(pp("B",tab));
}

```

```

tab.clear();
line.clear();
line={0,0,0,0};
tab.push_back(line);
tab.push_back(line);

line.clear();
line={1,1,1,1};
tab.push_back(line);
tab.push_back(line);

lit.insert(pp("~B",tab));
}

```

```

void D_form(vector<vector<bool>> &tab,vector<bool> &line)

```

```

{
line={0,0,0,0};
tab.push_back(line);
line={1,1,1,1};
for(int i=0;i<2;i++)
    tab.push_back(line);
line={0,0,0,0};
tab.push_back(line);
lit.insert(pp("D",tab));

tab.clear();

line={1,1,1,1};
tab.push_back(line);
line={0,0,0,0};
for(int i=0;i<2;i++)
    tab.push_back(line);
line={1,1,1,1};
tab.push_back(line);
lit.insert(pp("~D",tab));
}

```

```
}
```

```
void formare()
```

```
{
```

```
vector<vector<bool>> tab;
```

```
vector<bool> line;
```

```
string key1="A",key2="~A";
```

```
tab.clear();
```

```
line.clear();
```

```
line={1,1,0,0};
```

```
mtr1(tab,line);
```

```
lit.insert(pp("A",tab));
```

```
key1[0]++;
```

```
tab.clear();
```

```
line.clear();
```

```
line={0,0,1,1};
```

```
mtr1(tab,line);
```

```
lit.insert(pp("~A",tab));
```

```
key2[1]++;
```

```
tab.clear();
```

```
line.clear();
```

```
B_form(tab, line);
```

```
key1[0]++;
```

```
key2[1]++;
```

```
tab.clear();
```

```
line.clear();
```

```
line={0,1,1,0};
```

```
mtr1(tab,line);
lit.insert(pp("C",tab));
key1[0]++;
```

```
tab.clear();
line.clear();
line={1,0,0,1};
mtr1(tab,line);
lit.insert(pp("~C",tab));
key2[1]++;
tab.clear();
```

```
D_form(tab, line);
```

```
}
```

```
void multiplication(vector<vector<bool>> &a, vector<vector<bool>> b)
```

```
{
    for(int i=0;i<a.size();i++)
        for(int j=0;j<a[i].size();j++)
            a[i][j]= a[i][j]*b[i][j];
}
```

```
vector<vector<bool>> sol(string s)
```

```
{
    bool ok = 0;
    vector<vector<bool>> mat;
    for(int z=1;z<=4;z++)
        mat.push_back({1,1,1,1});
    int i=0, n=s.size();
    s+="QQQ";
    while (i<n)
    {
        if(s[i]!=s[i+1] and s[i]!=s[i+2] and s[i]!='~')
        {
```

```

ok = 1;
string key;
key.clear();
if(i>0)
{
    if(s[i-1]!='~')
        key+='~';
}
key+=s[i];
multiplication(mat, lit[key]);
i++;
}
else
{
    if(s[i]!='~')
        i++;
    else
    {
        if(s[i]==s[i+1])
            i+=2;
        else
            i+=3;
    }
}
}
if(!ok)
    mat = {{0,0,0,0},
           {0,0,0,0},
           {0,0,0,0},
           {0,0,0,0}};
return mat;
}

```

```

void print(vector<vector<bool>> a)
{

```

```

for(int i=0;i<a.size();i++)
{
    int sum=0;
    for(int j=0;j<a[i].size();j++)
        sum=sum+a[i][j]*pow(2, a[i].size()-1-j);
    cout<<cif[sum];
}
cout<<endl;
}

```

```

int main()
{
    formare();
    string exp;
    for(int i=1;i<=5;i++)
    {
        cin>>exp;
        exp+='+';
        vector<vector<bool>> rez;
        rez.clear();
        for(int z=1;z<=4;z++)
            rez.push_back({0,0,0,0});
        int j=0;
        string part;
        part.clear();
        while(j<exp.size())
        {
            if(exp[j]!='+')
            {
                part+=exp[j];
            }
            else
            {
                vector<vector<bool>> partial=sol(part);
            }
        }
    }
}

```

```
for(int z=0;z<partial.size();z++)
{
    for(int k=0;k<partial[z].size();k++)
        rez[z][k] = partial[z][k]+rez[z][k];
    }
    part.clear();
}
j++;
}
print(rez);

}
}
```