

```

//Bogdan Criveanu

#include <iostream>
#include <vector>
#include <string>
#include <algorithm>

using namespace std;

string s;
vector<int> next_poz = {0, 8, 8, 8, 8, 8, 8, 13, 13, 13, 13, 13, 18,
18, 18, 18, 23, 23, 23, 23, 28, 28, 28, 28, 28, 36, 36, 36, 36,
36, 36, 36, 36, 41, 41, 41, 41, 41, 46, 46, 46, 46, 46, 51, 51, 51, 51,
51, -1, -1, -1,};
vector<int> prime = {2, 3, 5, 7, 11, 13, 17, 19, 23, 29, 31, 37, 41,
43, 47, 53};
vector<int> pq = {9, 16, 25, 36, 49, 64};

bool verif(int newPoz, vector<int> op, vector<int> pl)
{
    if (newPoz > 52)
        return 1;
    for (auto &i : op)
        if (i == newPoz)
            return 1;
    for (auto &i : pl)
        if (i == newPoz)
            return 1;
    return 0;
}

int main()
{
    ios::sync_with_stdio(0);
    while (!cin.eof())
    {
        vector<int> op(3), pl(3);
        int n,d,newPoz,oldPoz;
        cin >> op[0] >> op[1] >> op[2] >> pl[0] >> pl[1] >> pl[2] >> n;
        if (cin.eof())
            break;
        sort(op.begin(), op.end());
        sort(pl.begin(), pl.end());
        for (int i = 0; i < n; i++)
        {
            cin >> d;
            newPoz = pl[0] + d;
            oldPoz = pl[0];
            pl[0] = 9999;
            int ok = 1;
            if (newPoz > 52)
                ok = 0;
            for (auto &p : op)
                if (p == newPoz)
                    ok = 0;
            for (auto &p : pl)

```

```

        if (p == newPoz)
            ok = 0;
if (!ok)
{
    pl[0] = oldPoz;
    continue;
}
if (newPoz == 52)
{
    vector<int> newPl;
    newPl.push_back(pl[1]);
    newPl.push_back(pl[2]);
    pl = newPl;
    sort(pl.begin(), pl.end());
    continue;
}
if (binary_search(prime.begin(), prime.end(), newPoz))
{
    for (int i = 1; i <= 6; i++, newPoz++)
        if (verif(newPoz + 1, op, pl))
            break;
    if (newPoz == 52)
    {
        vector<int> newPl;
        newPl.push_back(pl[1]);
        newPl.push_back(pl[2]);
        pl = newPl;
        sort(pl.begin(), pl.end());
        continue;
    }
    pl[0] = newPoz;
    sort(pl.begin(), pl.end());
    continue;
}
if (binary_search(pq.begin(), pq.end(), newPoz))
{
    for (int i = 1; i <= 6; i++, newPoz--)
        if (verif(newPoz - 1, op, pl))
            break;
    if (newPoz == 52)
    {
        vector<int> newPl;
        newPl.push_back(pl[1]);
        newPl.push_back(pl[2]);
        pl = newPl;
        sort(pl.begin(), pl.end());
        continue;
    }
    pl[0] = newPoz;
    sort(pl.begin(), pl.end());
    continue;
}
if (next_poz[oldPoz] > newPoz)
{
    pl[0] = newPoz;
    sort(pl.begin(), pl.end());
}

```

```

        continue;
    }
    if (verif(newPoz / d * d, op, pl))
    {
        pl[0] = oldPoz;
        continue;
    }

    newPoz = newPoz / d * d;
    if (newPoz == 52)
    {
        vector<int> newPl;
        newPl.push_back(pl[1]);
        newPl.push_back(pl[2]);
        pl = newPl;
        sort(pl.begin(), pl.end());
        continue;
    }
    pl[0] = newPoz;
    sort(pl.begin(), pl.end());
}
if (!pl.size())
    cout << "GAME OVER" << '\n';
else
{
    for (auto &p : pl)
        cout << p << " ";
    cout << "\n";
}
}
return 0;
}

```