```
//Radu Rotaru

#include <bits/stdc++.h>
using namespace std;
ifstream f("date.in");
ofstream g("date.out");
ofstream &operator<<(ofstream &os, vector<int> a)
{
    for (auto &i : a)
        os << i << " ";
    return os;
}
vector<int> solve(vector<int> &, vector<int> &, vector<int> &);
void move(int, int, vector<int> &, vector<int> &);
int getmin(vector<int> &);
bool eocup(int, vector<int> &, vector<int> &);
bool eprim(int &);
bool epp(int &);
bool schema(int, int);
int main()
{
    for (int teste = 5; teste; teste--)
    {
        vector<int> obs(3), ap(3), dice;
        int aux;
        for (int i = 0; i < 3; i++)
            f >> obs[i];
        for (int i = 0; i < 3; i++)
            f >> ap[i];
        f >> aux;
        dice.resize(aux);
        for (auto &i : dice)
            f >> i;
        auto af = solve(obs, ap, dice);
        if (af.empty())
            g << "GAME OVER\n";
        else
            g << af << '\n';
    }
    return 0;
}
vector<int> solve(vector<int> &obs, vector<int> &ap, vector<int> &dice)
{
    vector<int> rez;
    for (auto &i : dice)
        if (getmin(ap) != -1)
            move(getmin(ap), i, obs, ap);
    for (auto &i : ap)
        if (i != -1)
            rez.push_back(i);
    sort(rez.begin(), rez.end());
    return rez;
}
void move(int mn, int ct, vector<int> &obs, vector<int> &ap)
{
    int vp = ap[mn], newpoz = ap[mn] + ct;
```

```
    ap[mn] = -1;
    if (newpoz == 52)
        return;
    if (eocup(newpoz, obs, ap) || newpoz > 52)
    {
        ap[mn] = vp;
        return;
    }
    if (eprim(newpoz))
    {
        ap[mn] = newpoz;
        for (int i = newpoz + 1; i <= min(52, newpoz + 6); i++)
            if (!eocup(i, obs, ap))
                ap[mn] = i;
            else
                break;
        if (ap[mn] == 52)
            ap[mn] = -1;
    }
    else if (epp(newpoz))
    {
        ap[mn] = newpoz;
        for (int i = newpoz - 1; i >= max(1, newpoz - 6); i--)
            if (!eocup(i, obs, ap))
                ap[mn] = i;
            else
                break;
    }
    else if (schema(vp, newpoz))
    {
        ap[mn] = vp;
        for (int i = vp; i <= newpoz; i++)
            if (i % ct == 0 && !eocup(i, ap, obs))
                ap[mn] = i;
        if (ap[mn] == 52)
            ap[mn] = -1;
    }
    else
        ap[mn] = newpoz;
}
bool schema(int st, int dr)
{
    if (st <= 6)
        return (dr >= 8);
    if (st <= 11)
        return (dr >= 13);
    if (st <= 16)
        return dr >= 18;
    if (st <= 21)
        return dr >= 23;
    if (st <= 26)
        return dr >= 28;
    if (st <= 34)
        return dr >= 36;
    if (st <= 39)
        return dr >= 41;
```

```
    if (st <= 44)
        return dr <= 46;
    if (st <= 49)
        return dr >= 51;
    return 0;
}
bool eocup(int np, vector<int> &obs, vector<int> &ap)
{
    for (auto i : obs)
        if (i == np)
            return 1;
    for (auto i : ap)
        if (i == np)
            return 1;
    return 0;
}
int getmin(vector<int> &ap)
{
    int maxim = 999, ind = -1;
    for (int i = 0; i < 3; i++)
        if (ap[i] != -1 && ap[i] < maxim)
            maxim = ap[i], ind = i;
    return ind;
}
bool eprim(int &x)
{
    if (x == 2)
        return 1;
    if (x < 2 || x % 2 == 0)
        return 0;
    for (int i = 3; i * i <= x; i += 2)
        if (x % i == 0)
            return 0;
    return 1;
}
bool epp(int &x)
{
    if (x <= 4)
        return 0;
    return (sqrt(x) == int(sqrt(x)));
}
```