

```

/**
NAME: PINTILICIUC Ana
SCHOOL: NCSC "GRIGORE MOISIL"
GRADE: 11
DIVISION: INT5

**/

#include <fstream>
#include <iostream>
#include <queue>
#include <cmath>

using namespace std;

ifstream f("patolli.in");

struct marker
{
    int location;
    bool operator<(const marker &other) const
    {
        return location>other.location;
    }
};

int tabla[60];
int l1[9]={6, 11, 16, 21, 26, 34, 39, 44, 49};
int l2[9]={8, 13, 18, 23, 28, 36, 41, 46, 51};

priority_queue<marker>q;

bool prim(int x)
{
    if(x<2 || x>2 && x%2==0)
        return 0;
    for(int d=3; d*d<=x; d+=2)
        if(x%d==0)
            return 0;
    return 1;
}

marker moveMarker(int dice)
{
    marker Marker=q.top();
    q.pop();
    tabla[Marker.location]=0;
    Marker.location+=dice;
    if(tabla[Marker.location]==1)
    {
        Marker.location-=dice;
        tabla[Marker.location]=1;
        return Marker;
    }
}

```

```

    }
    if (Marker.location > 52)
    {
        Marker.location -= dice;
        tabla[Marker.location] = 1;
        return Marker;
    }
    if (prim(Marker.location) == 1)
    {
        int prevLocation = Marker.location;
        for (int i = prevLocation; i < prevLocation + 6 && tabla[i + 1] == 0; i++)
            Marker.location++;
        if (Marker.location > 52)
            Marker.location -= 6;
        tabla[Marker.location] = 1;
        return Marker;
    }
    if (sqrt(Marker.location) == (int) sqrt(Marker.location) &&
Marker.location > 4)
    {
        int prevLocation = Marker.location;
        for (int i = prevLocation; i > prevLocation - 6 && tabla[i - 1] == 0; i--)
            Marker.location--;
        tabla[Marker.location] = 1;
        return Marker;
    }
    int prevLocation = Marker.location - dice;
    int ok = 0;
    for (int v = 0; v < 9 && ok == 0; v++)
        if (prevLocation <= l1[v] && Marker.location >= l2[v])
            ok = 1;
    if (ok == 1)
    {
        Marker.location = prevLocation;
        for (int i = prevLocation; i <= prevLocation + dice; i++)
            if (i % dice == 0 && tabla[i] == 0)
                Marker.location = i;
    }
    tabla[Marker.location] = 1;
    return Marker;
}

void zero()
{
    for (int i = 1; i <= 55; i++)
        tabla[i] = 0;
}

int main()
{
    for (int acsl = 1; acsl <= 5; acsl++)
    {
        int x, moves;
        for (int i = 1; i <= 6; i++)

```

```

{
    f>>x;
    if(i>3)
    {
        marker aux;
        aux.location=x;
        q.push(aux);
    }
    tabla[x]=1;
}
f>>moves;
for(int i=1; i<=moves; i++)
{
    f>>x;
    q.push(moveMarker(x));
}

int ok=0;
while(!q.empty())
{
    if(q.top().location<52)
    {
        ok=1;
        cout<<q.top().location<<" ";
    }
    q.pop();
}
if(ok==0)
    cout<<"GAME OVER";
cout<<"\n";
zero();
}

return 0;
}

```