```
/**
NAME: Catrina Sorana
GRADE: 11
SCHOOL: NCSC "GR. MOISIL"
DIVISION: Int5
*/
#include <fstream>
#include <cmath>
#include <algorithm>
#include <queue>
using namespace std;

ifstream f("patolli.in");
ofstream g("patolli.out");

int grid[60];
int player[5];
int x;


int getmin(){
    int idx = 1;
    for(int i=1; i<=3; i++)
        if(player[i]<player[idx])
            idx = i;
    return idx;
}

int verif_vertical(int init, int fin){
    if(init<=6 && fin>=8)
        return 1;
    if(init<=11 && fin>=13)
        return 1;
    if(init<=16 && fin>=18)
        return 1;
    if(init <=21 && fin>=23)
        return 1;
    if(init <=26 && fin>=28)
        return 1;
    if(init<=34 && fin>=36)
        return 1;
    if(init <= 39 && fin>= 41)
        return 1;
    if(init <= 44 && fin>=46)
        return 1;
    if(init<=49 && fin>=51)
        return 1;
    return 0;
}

void zeroz(){
    for(int i=1; i<=60; i++)
        grid[i] = 0;
}
```

```
int prim(int x){
    if(x<2 || x%2 == 0 && x>2)
        return 0;
    for(int d=3; d*d<=x; d+=2)
        if(x%d == 0)
            return 0;
    return 1;
}
int main()
{
    for(int acsl=1; acsl<=5; acsl++){
        zeroz();
        for(int i=1; i<=3; i++){
            f>>x;
            grid[x] = 1;
        }
        for(int i=1; i<=3; i++){
            f>>x;
            player[i] = x;
            grid[x] = 1;
        }
        int k;
        int die;
        f>>k;
        for(int i=0; i<k; i++){
            f>>die;
            grid[52] = 0;
            int pozdemutat = getmin();
            int valdemut = player[pozdemutat];
            if(grid[valdemut + die] == 1)
                continue;
            if(valdemut + die > 52)
                continue;
            if(valdemut + die == 52)
                grid[pozdemutat] = 65;
            if(prim(valdemut + die) == 1){
                int j;
                grid[valdemut] = 0;
                for(j=valdemut+die; grid[j]!=1 && j-valdemut-die<=6;
j++);
                j--;
                player[pozdemutat] = j;
                grid[j] = 1;
                continue;
            }
            else if(sqrt(valdemut+die)==(int)(sqrt(valdemut+die)) &&
valdemut + die > 4){
                int j;
                grid[valdemut] = 0;
                for(j=valdemut+die; grid[j]!=1 && valdemut+die-j<=6; j--
);
                j++;
                player[pozdemutat] = j;
                grid[j] = 1;
```

```
            }
            else if(verif_vertical(valdemut, valdemut+die) == 1){
                int deunde = (valdemut+die)/die*die;
                int poz = valdemut;
                for(int j = deunde; j>=valdemut; j-=die)
                    if(grid[j] == 0){
                        poz = j;
                        break;
                    }
                grid[valdemut] = 0;
                grid[poz] = 1;
                player[pozdemutat] = poz;
            }
            else{
                grid[valdemut] = 0;
                grid[valdemut + die] = 1;
                player[pozdemutat] = valdemut + die;
            }
        }
        sort(player+1, player+4);
        int ok=0;
        for(int i=1; i<=3; i++)
            if(player[i] < 52)
            {
                ok = 1;
                g<<player[i]<<" ";
            }
        if(ok == 0)
            g<<"GAME OVER";
        g<<'\n';

    }
    return 0;
}
```