

```
/**
    NAME: Comaniciu Maria
    SCHOOL: NCSC "Grigore Moisil"
    GRADE: 11
    DIVISION: SR3
**/
```

```
#include <fstream>
#include <iostream>
#include <algorithm>
```

```
using namespace std;
```

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

```
int prime[55], patrate[55];
int game[4][5];
//int player[5];
//int opponent[5];
int rolls, howMuch;
int lin[55], col[55];
int grid[13][13];
int nrOnGrid[4];
int turn;
int a[13][13];
```

```
pair <int, int> unde(int x)
{
    if(x == 1)
        return {1, 5};
    if(x == 2)
        return {2, 5};
    if(x>=3 && x<=7)
        return {3, 7-x+1};
    if(x>=8 && x<=12)
        return {4, x-8+1};
    if(x>=13 && x<=17)
        return {5, 17-x+1};
    if(x>=18 && x<=22)
        return {6, x-18+1};
    if(x>=23 && x<=26)
        return {7+x-23, 5};
    if(x>=27 && x<=30)
        return {7+30-x, 6};
    if(x>=31 && x<=35)
        return {6, 6+x-31};
    if(x>=36 && x<=40)
        return {5, 6+40-x};
    if(x>=41 && x<=45)
        return {4, 6+x-41};
    if(x>=46 && x<=50)
        return {3, 6+50-x};
    if(x == 51)
```

```

        return {2, 6};
    if(x == 52)
        return {1, 6};
    return {1, 1};
}

void pun()
{
    for(int i=1; i<=52; i++)
    {
        auto poz=unde(i);
        lin[i]=poz.first;
        col[i]=poz.second;
    }
}

void afis()
{
    for(int i=1; i<=10; i++)
    {
        for(int j=1; j<=10; j++)
            cout<<a[i][j]<<' ';
        cout<<'\n';
    }
    cout<<'\n';
}

void ciur()
{
    prime[1]=1;
    for(int i=2; i<=53; i++)
    {
        if(prime[i] == 0)
        {
            for(int j=i+i; j<=53; j+=i)
                prime[j]=1;
        }
    }
}

void squares()
{
    for(int i=3; i*i<=53; i++)
    {
        patrate[i*i]=1;
    }
}

int mutiUp(int poz)
{
    int linaici;
    int colaici;
    for(int i=poz+1; i<=poz+6; i++)
    {
        linaici=lin[i];
    }
}

```

```

        colaici=col[i];
        if(a[linaici][colaici] == 1)
            return i-1;
        if(i > 52)
            return poz;
    }
    return poz+6;
}

int mutiDown(int poz)
{
    int linaici;
    int colaici;
    for(int i=poz-1; i>=poz-6; i--)
    {
        linaici=lin[i];
        colaici=col[i];
        if(a[linaici][colaici] == 1)
            return i+1;
    }
    return poz-6;
}

int faParcurgerea(int pozin, int pozfin)
{
    int hor=0;
    int lina, cola, linant=lin[pozin], colant=col[pozin];
    for(int i=pozin+1; i<=pozfin; i++)
    {
        lina=lin[i];
        cola=col[i];
        if(cola == colant-1 || cola == colant+1)
            hor=1;
        else if(hor == 1 && (lina == linant-1 || lina == linant+1))
            return 1;
        linant=lina;
        colant=cola;
    }
    return 0;
}

int mutaReg9(int pozin, int pozfin, int dice)
{
    for(int i=pozfin; i>=pozin; i--)
    {
        int loclin=lin[i];
        int loccol=col[i];
        if(a[loclin][loccol] == 0 && i%dice == 0)
            return i;
    }
    return pozin;
}

void unPas(int val, int turn)

```

```

{
    sort(game[turn]+1, game[turn]+1+nrOnGrid[turn]);
    int newEl=game[turn][1]+val;
    int acumLin=lin[game[turn][1]];
    int acumCol=col[game[turn][1]];
    a[acumLin][acumCol]=0;
    if(a[lin[newEl]][col[newEl]] == 1)
        return;
    if(newEl == 52)
    {
        a[acumLin][acumCol]=0;
        game[turn][1]=70;
        sort(game[turn]+1, game[turn]+1+nrOnGrid[turn]);
        nrOnGrid[turn]--;
        return;
    }
    if(newEl>52)
        return;
    if(prime[newEl] == 0)
    {
        int nouPoz=mutiUp(newEl);
        a[acumLin][acumCol]=0;
        game[turn][1]=nouPoz;
        a[lin[nouPoz]][col[nouPoz]]=1;
        return;
    }
    if(patrate[newEl] == 1)
    {
        int nouPoz=mutiDown(newEl);
        a[acumLin][acumCol]=0;
        game[turn][1]=nouPoz;
        a[lin[nouPoz]][col[nouPoz]]=1;
        return;
    }
    if(faParcurerea(game[turn][1], newEl))
    {
        int nouPoz=mutaReg9(game[turn][1], newEl, val);
        a[acumLin][acumCol]=0;
        game[turn][1]=nouPoz;
        a[lin[nouPoz]][col[nouPoz]]=1;
        return;
    }
    a[acumLin][acumCol]=0;
    game[turn][1]=newEl;
    a[lin[newEl]][col[newEl]]=1;
}

void zero()
{
    for(int i=1; i<=10; i++)
        for(int j=1; j<=10; j++)
            a[i][j]=0;
}

```

```

int main()
{
    ciur();
    squares();
    pun();
    for(int acsl=1; acsl<=5; acsl++)
    {
        nrOnGrid[1]=3;
        nrOnGrid[2]=3;
        zero();
        turn=1;
        //afis();
        for(int i=1; i<=3; i++)
        {
            f>>game[turn][i];
            a[lin[game[turn][i]]][col[game[turn][i]]]=1;
        }
        sort(game[turn]+1, game[turn]+4);
        //afis();
        turn=2;
        //afis();
        for(int i=1; i<=3; i++)
        {
            f>>game[turn][i];
            a[lin[game[turn][i]]][col[game[turn][i]]]=1;
        }
        //afis();
        f>>rolls;
        for(int roll=1; roll<=rolls; roll++)
        {
            f>>howMuch;
            turn=turn%2+1;
            unPas(howMuch, turn);
            //afis();
        }
        sort(game[1]+1, game[1]+nrOnGrid[1]+1);
        sort(game[2]+1, game[2]+nrOnGrid[2]+1);
        int sum1=0, sum2=0;
        for(int i=1; i<=nrOnGrid[1]; i++)
            sum1=sum1+game[1][i];
        for(int i=1; i<=nrOnGrid[2]; i++)
            sum2=sum2+game[2][i];
        cout<<sum1<<' '<<sum2<<'\n';
    }

    return 0;
}

```