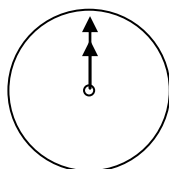


Bài: NHỮNG CHIẾC ĐỒNG HỒ

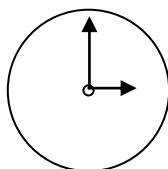
Có 9 chiếc đồng hồ được sắp thành ma trận 3x3 như sau:

A B C
D E F
G H I

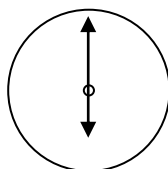
Mỗi chiếc đồng hồ có 4 trạng thái là:



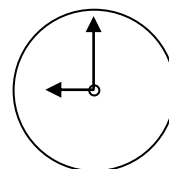
Trạng thái 0
12h



Trạng thái 1
3h



Trạng thái 2
6h



Trạng thái 3
9h

Cho 9 phép biến đổi. Mỗi phép biến đổi là một phép tác dụng lên một số đồng hồ nhất định, tăng thêm 3h của đồng hồ (thay đổi trạng thái đồng hồ):

- | | | |
|----|------------------|---------------|
| 1: | thay đổi đồng hồ | A, B, D, E |
| 2: | | A, B, C |
| 3: | | B, C, E, F |
| 4: | | A, D, G |
| 5: | | B, D, E, F, H |
| 6: | | C, F, I |
| 7: | | D, E, G, H |
| 8: | | G, H, I |
| 9: | | E, F, H, I |

Cho trước các trạng thái của 9 đồng hồ. Hãy thực hiện một số ít nhất các phép biến đổi nói trên để đưa các đồng hồ về trạng thái 0.

Ví dụ:

Trạng thái ban đầu:

0 2 3
2 1 2
3 2 0

Ta chỉ sử dụng 3 lần biến đổi theo tự là 3, 5 và 7

0 2 3		0 3 0		0 0 0		0 0 0
2 1 2	→3→	2 2 3	→5→	3 3 0	→7→	0 0 0
3 2 0		3 2 0		3 3 0		0 0 0

Code Pascal

```

const FileInput='CLOCKS.INP';
      FileOutPut='CLOCK.OUT';
      Change: array[1..9,1..9] of byte =
((1,1,0,1,1,0,0,0,0),

(1,1,1,0,0,0,0,0,0),

(0,1,1,0,1,1,0,0,0),

(1,0,0,1,0,0,1,0,0),

(0,1,0,1,1,1,0,1,0),

(0,0,1,0,0,1,0,0,1),

(0,0,0,1,1,0,1,1,0),

(0,0,0,0,0,0,1,1,1),

(0,0,0,0,1,1,0,1,1));
      maxC = 30;

var Clocks: array[1..9] of byte;
    Solution, minSolution: array[1..9] of byte;
    found: boolean;
    min: byte;

procedure Init;
var fi: text;
    i, j: byte;
begin
    assign(fi,FileInput); reset(fi);
    for i:=1 to 3 do
        begin
            for j:=1 to 3 do read(fi,clocks[(i-1)*3 + j]);
            readln(fi);
        end;
    close(fi);
    min:=maxC;
    fillchar(minSolution, sizeof(minSolution), 0);
    found:= false;
end;

procedure CheckSolution;
var i, j: byte;
    a: array[1..9] of byte;
begin
    for i:=1 to 9 do a[i]:=Clocks[i];
    for i:=1 to 9 do

```

```

        for j:=1 to 9 do
            if Change[i,j]=1 then a[j]:= (a[j]+Solution[i]) mod
4;
        found:= true;
        for i:=1 to 9 do if a[i]>0 then found:=false;
    end;

procedure OptimizeSolution;
var i, s: byte;
begin
    s:=0;
    for i:=1 to 9 do s:=s+Solution[i];
    if min>s then
        begin
            min:=s;
            for i:=1 to 9 do minSolution[i]:=Solution[i];
        end;
    end;

procedure FindSolution(i: byte);
var j: byte;
begin
    for j:=0 to 3 do
        begin
            solution[i]:=j;
            if i=9 then CheckSolution
            else FindSolution(i+1);
            if found then OptimizeSolution;
        end;
    end;

procedure Output;
var i, j: byte;
    fo: text;
begin
    assign(fo, FileOutput); rewrite(fo);
    if min=maxC then min:=0;
    writeln(fo,min);
    for i:=1 to 9 do
        for j:=1 to minSolution[i] do write(fo,i,' ');
    close(fo);
end;

BEGIN
    Init;
    FindSolution(1);
    Output;
END.

```