

B. 2048

PROBLEM DESCRIPTION

Simulate a single move of the viral game: 2048.

SOLUTION TECHINQUES

Simulation

SOLUTION SKETCHES

There are many approaches to this type of simulation.
One can use `std::deque` just as solution program did.

TIME COMPLEXITY

$O(1)$ with a slightly big constant.

SOLUTION PROGRAM FOR REFERENCE (AUTHOR: PWLIAO)

```
1. #include <cstdio>
2. #include <deque>
3. #include <algorithm>
4. using namespace std;
5. int g[4][4],dir;
6. // left up right down
7. int r[4];
8. void sol(deque<int> &q)
9. {
10.     for(int i=q.size();i<4;i++)
11.         q.push_back(0);
12.     for(int i=0;i<4;i++)
13.         r[i]=0;
14.     if(q[0]==q[1])
15.     {
16.         r[0]=q[0]+q[1];
17.         if(q[2]==q[3])
18.         {
19.             r[1]=q[2]+q[3];
20.         }
21.         else
22.         {
23.             r[1]=q[2];
24.             r[2]=q[3];
25.         }
26.     }
27.     else if(q[1]==q[2])
28.     {
29.         r[0]=q[0];
30.         r[1]=q[1]+q[2];
31.         r[2]=q[3];
32.     }
33.     else if(q[2]==q[3])
34.     {
35.         r[0]=q[0],r[1]=q[1];
36.         r[2]=q[2]+q[3];
37.     }
38.     else
39.     {
40.         for(int i=0;i<4;i++)
41.         {
42.             r[i]=q[i];
43.         }
44.     }
45. }
46. int main()
47. {
48.     for(int i=0;i<4;i++)
49.         for(int j=0;j<4;j++)
50.             scanf("%d",&g[i][j]);
51.     scanf("%d",&dir);
52.     if(dir==0)
53.     {
54.         for(int i=0;i<4;i++)
55.         {
56.             deque<int> q;
57.             for(int j=0;j<4;j++)
```

```

58.         {
59.             if(g[i][j])
60.                 q.push_back(g[i][j]);
61.         }
62.         sol(q);
63.         for(int j=0;j<4;j++)
64.             g[i][j]=r[j];
65.     }
66. }
67. else if(dir==1)
68. {
69.     for(int i=0;i<4;i++)
70.     {
71.         deque<int> q;
72.         for(int j=0;j<4;j++)
73.         {
74.             if(g[j][i])
75.                 q.push_back(g[j][i]);
76.         }
77.         sol(q);
78.         for(int j=0;j<4;j++)
79.             g[j][i]=r[j];
80.     }
81. }
82. else if(dir==2)
83. {
84.     for(int i=0;i<4;i++)
85.     {
86.         deque<int> q;
87.         for(int j=3;j>=0;j--)
88.         {
89.             if(g[i][j])
90.                 q.push_back(g[i][j]);
91.         }
92.         sol(q);
93.         for(int j=3;j>=0;j--)
94.             g[i][j]=r[3-j];
95.     }
96. }
97. else
98. {
99.     for(int i=0;i<4;i++)
100.    {
101.        deque<int> q;
102.        for(int j=3;j>=0;j--)
103.        {
104.            if(g[j][i])
105.                q.push_back(g[j][i]);
106.        }
107.        sol(q);
108.        for(int j=3;j>=0;j--)
109.            g[j][i]=r[3-j];
110.    }
111. }
112. for(int i=0;i<4;i++)
113. {
114.     for(int j=0;j<4;j++)
115.     {
116.         if(j) putchar(' ');
117.         printf("%d",g[i][j]);
118.     }
119.     puts("");
120. }

```

```
119.         return 0;  
120.     }
```