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)

```
#include <cstdio>
#include <deque>
#include <algorithm>
using namespace std;
int g[4][4],dir;
// left up right down
int r[4];
void sol(deque<int> &q)
{
    for(int i=q.size();i<4;i++)
        q.push_back(0);
    for(int i=0;i<4;i++)
        r[i]=0;
    if(q[0]==q[1])
    {
        r[0]=q[0]+q[1];
        if(q[2]==q[3])
        {
            r[1]=q[2]+q[3];
        }
        else
        {
            r[1]=q[2];
            r[2]=q[3];
        }
    }
    else if(q[1]==q[2])
    {
        r[0]=q[0];
        r[1]=q[1]+q[2];
        r[2]=q[3];
    }
    else if(q[2]==q[3])
    {
        r[0]=q[0],r[1]=q[1];
        r[2]=q[2]+q[3];
    }
    else
    {
        for(int i=0;i<4;i++)
        {
            r[i]=q[i];
        }
    }
}
int main()
{
    for(int i=0;i<4;i++)
        for(int j=0;j<4;j++)
            scanf("%d",&g[i][j]);
    scanf("%d",&dir);
    if(dir==0)
    {
        for(int i=0;i<4;i++)
        {
            deque<int> q;
            for(int j=0;j<4;j++)
```

```
            {
                if(g[i][j])
                    q.push_back(g[i][j]);
            }
            sol(q);
            for(int j=0;j<4;j++)
                g[i][j]=r[j];
        }
    }
    else if(dir==1)
    {
        for(int i=0;i<4;i++)
        {
            deque<int> q;
            for(int j=0;j<4;j++)
            {
                if(g[j][i])
                    q.push_back(g[j][i]);
            }
            sol(q);
            for(int j=0;j<4;j++)
                g[j][i]=r[j];
        }
    }
    else if(dir==2)
    {
        for(int i=0;i<4;i++)
        {
            deque<int> q;
            for(int j=3;j>=0;j--)
            {
                if(g[i][j])
                    q.push_back(g[i][j]);
            }
            sol(q);
            for(int j=3;j>=0;j--)
                g[i][j]=r[3-j];
        }
    }
    else
    {
        for(int i=0;i<4;i++)
            {
                deque<int> q;
                for(int j=3;j>=0;j--)
                    if(g[j][i])
                        q.push_back(g[j][i]);
                sol(q);
                for(int j=3;j>=0;j--)
                    g[j][i]=r[3-j];
            }
         }
         for(int i=0;i<4;i++)
         {
             for(int j=0;j<4;j++)
             {
                 if(j) putchar(' ');
                 printf("%d",g[i][j]);
             }
             puts("");
         }
```

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