经典屏保题目描述：

DVD 机在视频输出时，为了保护电视显像管，在待机状态会显示“屏保动画”，如下图所示，DVD Logo 在屏幕内来回运动，碰到边缘会反弹：

请根据如下要求，实现屏保 Logo 坐标的计算算法。

1、屏幕是一个 800*600 像素的矩形，规定屏幕的左上角点坐标原点，沿横边向右方向为 X 轴，沿竖边向下方向为 Y 轴；

2、Logo 是一个 50*25 像素的矩形，初始状态下，左上角点坐标记做（x，y），它在 X 和 Y 方向上均以 1 像素/秒的速度开始运动；

3、遇到屏幕四个边缘后，会发生镜面反弹，即以 45°碰撞边缘，再改变方向以 45°弹出；

4、当 Logo 和四个角碰撞时，两个边缘同时反弹的效果是 Logo 会原路返回。

请编码实现，t 秒后 Logo 左上角点的坐标。

输入描述：

输入 3 个数字，以空格分隔：

x y t

第一个数字表示 Logo 左上角点的初始 X 坐标；

第二个数字表示 Logo 左上角点的初始 Y 坐标；

第三个数字表示时间 t，题目要求即求 t 秒后 Logo 左上角点的位置。

输出描述：

输出 2 个数字，以空格分隔：

x y

第一个数字表示 t 秒后，Logo 左上角点的 X 坐标

第二个数字表示 t 秒后，Logo 左上角点的 Y 坐标

补充说明：
所有用例均保证：

1、输入的 x 和 y 坐标会保证整个 Logo 都在屏幕范围内，Logo 不会出画；

2、所有输入数据都是合法的数值，且不会出现负数；

3、t 的最大值为 100000。

示例 1

输入：

```
0 0 10
```

输出：

```
10 10
```

说明：

输入样例表示 Logo 初始位置在屏幕的左上角点，10s 后，Logo 在 X 和 Y 方向都移动了 10 像素，因此输出 10 10。

示例 2

输入：

```
500 570 10
```

输出：

```
510 570
```

说明：

输入样例表示初始状态下，Logo 的下边缘再有 5 像素就碰到屏幕下边缘了，5s 后，会与屏幕碰撞，碰撞后，斜向 45°弹出，又经过 5s 后，Logo 与起始位置相比，水平移动了 10 像素，垂直方向回到了原来的高度。

```
#include <stdarg.h>

#include <iostream>

#include <vector>

#include <limits>

#include <unordered_map>

#include <unordered_set>

#include <algorithm>

#include <queue>

#include <array>

#include <numeric>

using namespace std;


int main() {
```

```
struct point{int x, y;};


enum class direction{
    UP, DOWN, LEFT, RIGHT,
    UP_RIGHT, UP_LEFT,
    DOWN_RIGHT, DOWN_LEFT
};


point pos;
int time;
cin >> pos.x >> pos.y >> time;
direction cur_direct = direction::DOWN_RIGHT;


int cur_time = 0;
while (cur_time < time)
{
    switch (cur_direct)
    {
    case direction::DOWN_RIGHT:{
        int x_time = 799 - (pos.x + 49);
        int y_time = 599 - (pos.y + 24);
        if (cur_time + min(x_time, y_time) >= time){
```

```
        pos.x += time - cur_time;
        pos.y += time - cur_time;
        cur_time = time;
        break; // 要跳出 while
    }
    if (x_time == y_time){
        pos.x += x_time;
        pos.y += x_time;
        cur_time += x_time;
        cur_direct = direction::UP_LEFT;
        break;
    }
    else if (x_time < y_time){  // 碰到右边
        pos.x += x_time;
        pos.y += x_time;
        cur_time += x_time;
        cur_direct = direction::DOWN_LEFT;
        break;
    }
    else{              // 碰到下面
        pos.x += y_time;
        pos.y += y_time;
```

```
            cur_time += y_time;
            cur_direct = direction::UP_RIGHT;
        }
        break;
    }
    case direction::DOWN_LEFT:{
        int x_time = pos.x;
        int y_time = 599 - (pos.y + 24);

        if (cur_time + min(x_time, y_time) >= time){
            pos.x -= time - cur_time;
            pos.y += time - cur_time;
            cur_time = time;
            break; // 要跳出 while
        }
        if (x_time == y_time){
            pos.x -= x_time;
            pos.y += x_time;
            cur_time += x_time;
            cur_direct = direction::UP_RIGHT;
            break;
        }
```

```
        else if (x_time < y_time){        // 碰到左边
            pos.x -= x_time;
            pos.y += x_time;
            cur_time += x_time;
            cur_direct = direction::DOWN_RIGHT;
            break;
        }
        else{          // 碰到下面
            pos.x -= y_time;
            pos.y += y_time;
            cur_time += y_time;
            cur_direct = direction::UP_LEFT;
        }
        break;
    }

    case direction::UP_LEFT:{
        int x_time = pos.x;
        int y_time = pos.y;

        if (cur_time + min(x_time, y_time) >= time){
            pos.x -= time - cur_time;
```

```
        pos.y -= time - cur_time;
        cur_time = time;
        break; // 要跳出 while
    }
    if (x_time == y_time){
        pos.x -= x_time;
        pos.y -= x_time;
        cur_time += x_time;
        cur_direct = direction::DOWN_RIGHT;
        break;
    }
    else if (x_time < y_time){  // 碰到左边
        pos.x -= x_time;
        pos.y -= x_time;
        cur_time += x_time;
        cur_direct = direction::UP_RIGHT;
        break;
    }
    else{   // 碰到上面
        pos.x -= y_time;
        pos.y -= y_time;
        cur_time += y_time;
```

```
            cur_direct = direction::DOWN_LEFT;
        }
        break;
    }


    case direction::UP_RIGHT:{
        int x_time = 799 - (pos.x + 49);
        int y_time = pos.y;

        if (cur_time + min(x_time, y_time) >= time){
            pos.x += time - cur_time;
            pos.y -= time - cur_time;
            cur_time = time;
            break; // 要跳出 while
        }
        if (x_time == y_time){
            pos.x += x_time;
            pos.y -= x_time;
            cur_time += x_time;
            cur_direct = direction::DOWN_LEFT;
            break;
```

```
            }
            else if (x_time < y_time){        // 碰到右边
                pos.x += x_time;
                pos.y -= x_time;
                cur_time += x_time;
                cur_direct = direction::UP_LEFT;
                break;
            }
            else{          // 碰到上面
                pos.x += y_time;
                pos.y -= y_time;
                cur_time += y_time;
                cur_direct = direction::DOWN_RIGHT;
            }
            break;
        }
        }
    }

    cout << pos.x << " " << pos.y << endl;

}
```