```cpp
#include<bits/stdc++.h>
using namespace std;

#define MAXN 1005

int dx[] = {1, 1, 1, -1, -1, -1, 0, 0};
int dy[] = {1, -1, 0, 1, -1, 0, 1, -1};
int n, m;
int grid[MAXN][MAXN];
vector<pair<int, int>> path[MAXN];
int vis[MAXN][MAXN];

struct Drone {
    int x, y, fx, fy, t, id;
    bool operator<(const Drone& other) const {
        return t < other.t;
    }
};

vector<Drone> drones;

void bfs(Drone drone) {
    memset(vis, 0, sizeof(vis));
    queue<pair<int, int>> q;
    q.push({drone.x, drone.y});
    vis[drone.x][drone.y] = 1;
    while(!q.empty()) {
        pair<int, int> curr = q.front();
        q.pop();
        for(int i = 0; i < 8; i++) {
            int x = curr.first + dx[i];
            int y = curr.second + dy[i];
            if(x >= 0 && x < n && y >= 0 && y < m && !vis[
                vis[x][y] = 1;
```

```
35:                 q.push({x, y});
36:                 path[drone.id].push_back({x, y});
37:                 if(x == drone.fx && y == drone.fy) {
38:                     return;
39:                 }
40:             }
41:         }
42:     }
43: }
44:
45: int main() {
46:     memset(grid, -1, sizeof(grid));
47:     cin >> n >> m;
48:     int id = 0;
49:     while(true) {
50:         int x1, y1, x2, y2, t;
51:         cin >> x1 >> y1 >> x2 >> y2 >> t;
52:         if(cin.fail()) {
53:             break;
54:         }
55:         Drone drone;
56:         drone.x = x1, drone.y = y1, drone.fx = x2, drone.f
57:         drones.push_back(drone);
58:     }
59:     sort(drones.begin(), drones.end());
60:     for(int i = 0; i < drones.size(); i++) {
61:         Drone drone = drones[i];
62:         bfs(drone);
63:         for(int j = 0; j < path[drone.id].size(); j++) {
64:             pair<int, int> curr = path[drone.id][j];
65:             grid[curr.first][curr.second] = drone.id;
66:         }
67:     }
68:     for(int i = 0; i < drones.size(); i++) {
```

```
69:         Drone drone = drones[i];
70:         cout << "Drone " << drone.id + 1 << ": ";
71:         for(int j = 0; j < path[drone.id].size(); j++) {
72:             pair<int, int> curr = path[drone.id][j];
73:             cout << "(" << curr.first << ", " << curr.seco
74:         }
75:         cout << endl;
76:     }
77:     return 0;
78: }
```