```c
#include <stdio.h>

// Are (a,b) valid indices in a p x q 2-D array?
int valid(int a, int b, int p, int q){
    if(a < 0 || b < 0)
        return 0;
    if(a >= p || b >= q)
        return 0;
    return 1;
}

int min(int a, int b){
    return a < b ? a : b;
}

int main(){
    int M, N, P;
    // Illegal value to tell us that the city does not have a PO
    int illegal = 100000000;

    scanf("%d %d %d", &M, &N, &P);

    int town[M][N]; // Which post office services this town
    int post[M][N]; // Locations of the various PO
    int coun[P][2]; // Number of towns serviced by various PO

    for(int i=0; i<M; i++){
        for(int j=0; j<N; j++){
            post[i][j] = illegal;
            town[i][j] = illegal;
        }
    }

    for(int i=0; i<P; i++){
        int a, b, p;
        scanf("%d %d %d", &a, &b, &p);
        post[a-1][b-1] = p;
        town[a-1][b-1] = p;
        coun[i][0] = p; // Name of this PO
        coun[i][1] = 0; // Right now this PO serves none - count later
    }

    for(int i=0; i<M; i++){
        for(int j=0; j<N; j++){
            // Hmm ... this town does not have a PO
            // Find a PO to service this town
            if(town[i][j] == illegal){
                if(valid(i-1, j-1, M, N))
                    town[i][j] = min(town[i][j], post[i-1][j-1]);
                if(valid(i-1, j  , M, N))
                    town[i][j] = min(town[i][j], post[i-1][j]);
                if(valid(i-1, j+1, M, N))
                    town[i][j] = min(town[i][j], post[i-1][j+1]);
                if(valid(i  , j-1, M, N))
                    town[i][j] = min(town[i][j], post[i][j-1]);
                if(valid(i  , j+1, M, N))
                    town[i][j] = min(town[i][j], post[i][j+1]);
                if(valid(i+1, j-1, M, N))
                    town[i][j] = min(town[i][j], post[i+1][j-1]);
                if(valid(i+1, j  , M, N))
                    town[i][j] = min(town[i][j], post[i+1][j]);
                if(valid(i+1, j+1, M, N))
                    town[i][j] = min(town[i][j], post[i+1][j+1]);
            }
        }
    }

    int ns = 0; // Number of towns not serviced at all!
    for(int i=0; i<M; i++){
```

```
        for(int j=0; j<N; j++){
            ns += (town[i][j] == illegal);
        }
    }

    // Find, for each PO, how many towns it serves
    for(int i = 0; i < M; i++){
        for(int j = 0; j < N; j++){
            int p = town[i][j];
            if(p != illegal) // PO p serves this town
                for(int k = 0; k < P; k++)
                    if(coun[k][0] == p)
                        coun[k][1]++; // Give credit to that PO
        }
    }

    int val = -1, pin; // Busiest PO
    for(int i=0; i<P; i++){
        if(coun[i][1] > val){
            val = coun[i][1];
            pin = coun[i][0];
        }
        else if(coun[i][1] == val){
            pin = min(pin, coun[i][0]);
        }
    }

    printf("NO PO: %d\nBUSIEST PO: %d", ns, pin);
    return 0;
}
```