```cpp
#include <cstdio>
#include <cstring>
#include <cstdlib>

const int N_MAX = 111;
const int INF = 0x3f3f3f3f;

int L[N_MAX * N_MAX], R[N_MAX * N_MAX],
    U[N_MAX * N_MAX], D[N_MAX * N_MAX], C[N_MAX * N_MAX];
int S[N_MAX], O[N_MAX];

int mat[N_MAX][N_MAX];
int match[N_MAX][N_MAX];

struct Coord{
    int x, y;
} pos[N_MAX];

void cover(int c){
    L[R[c]] = L[c], R[L[c]] = R[c];
    for(int i = D[c]; i != c; i = D[i])
        for(int j = R[i]; j != i; j = R[j])
            U[D[j]] = U[j], D[U[j]] = D[j], S[C[j]] --;
}

void uncover(int c){
    for(int i = U[c]; i != c; i = U[i])
        for(int j = L[i]; j != i; j = L[j])
            S[C[j]] ++, U[D[j]] = j, D[U[j]] = j;
    L[R[c]] = c, R[L[c]] = c;
}

void search(int k){
    if(R[0] == 0){
        for(int i = 0; i < k; i ++)
            printf("%d\n", pos[O[i]].x);
        exit(0);
    }
    int s = INF, c;
    for(int j = R[0]; j != 0; j = R[j])
        if(S[j] < s)
            c = j, s = S[j];
    cover(c);
    for(int r = D[c]; r != c; r = D[r]){
        O[k] = r;
        for(int j = R[r]; j != r; j = R[j])
```

```
            cover(C[j]);
        search(k + 1);
        for(int j = L[r]; j != r; j = L[j])
            uncover(C[j]);
    }
    uncover(c);
}

void add(int x, int y, int &cnt){
    match[x][y] = cnt;
    pos[cnt].x = x, pos[cnt].y = y;
    cnt ++;
}

void build(int mat[N_MAX][N_MAX], int n, int m){
    memset(match, -1, sizeof(match));
    int cnt = 0;
    add(0, 0, cnt);
    for(int j = 1; j <= m; j ++)
        add(0, j, cnt);
    for(int i = 1; i <= n; i ++)
        for(int j = 1; j <= m; j ++)
            if(mat[i][j] == 1)
                add(i, j, cnt);
    for(int i = 0; i <= n; i ++){
        static int tmp[N_MAX];
        int n_tmp = 0;
        for(int j = 0; j <= m; j ++)
            if(match[i][j] != -1)
                tmp[n_tmp ++] = match[i][j];
        tmp[n_tmp] = tmp[0];
        for(int k = 0; k < n_tmp; k ++)
            R[tmp[k]] = tmp[k + 1];
        for(int k = 1; k <= n_tmp; k ++)
            L[tmp[k]] = tmp[k - 1];
    }
    for(int j = 0; j <= m; j ++){
        static int tmp[N_MAX];
        int n_tmp = 0;
        for(int i = 0; i <= n; i ++)
            if(match[i][j] != -1)
                tmp[n_tmp ++] = match[i][j], C[match[i][j]] = match[0][j];
        tmp[n_tmp] = tmp[0];
        for(int k = 0; k < n_tmp; k ++)
            D[tmp[k]] = tmp[k + 1];
        for(int k = 1; k <= n_tmp; k ++)
```

```
            U[tmp[k]] = tmp[k - 1];
        S[j] = n_tmp;
    }
}

int main(){
    freopen("t.in", "r", stdin);
    int n, m;
    scanf("%d%d", &n, &m);
    for(int i = 1; i <= n; i ++)
        for(int j = 1; j <= m; j ++)
            scanf("%d", &mat[i][j]);
    build(mat, n, m);
    search(0);
}
```