```c
#include <stdio.h>
#include <stdlib.h>

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

int isIdentity(int **arr, int ox, int oy, int size){
    int i, j, isIden = 1;
    for(i = 0; i < size; i++){
        for(j = 0; j < size; j++){
            if(i == j){
                if(arr[ox + i][oy + j] != 1)
                    isIden = 0;
            }else{
                if(arr[ox + i][oy + j] != 0)
                    isIden = 0;
            }
        }
    }
    return isIden;
}

int main() {
    int n, m, i, j, k, kMax;
    int size = 0, idx1 = -1, idx2 = -1;
    scanf("%d %d", &n, &m);
    int **arr = (int**)malloc(n * sizeof(int*));

    for(i = 0; i < n; i++){
        arr[i] = (int*)malloc(m * sizeof(int));
        for(j = 0; j < m; j++)
            scanf("%d", &arr[i][j]);
    }

    for(i = 0; i < n; i++){
        for(j = 0; j < m; j++){
            // How far are the boundaries of the matrix
            // That is the size of identity matrix that can fit in
            kMax = min(n - i, m - j);
            for(k = 0; k < kMax; k++){
                // Is the k x k matrix starting at (i,j) an identity?
                if(isIdentity(arr, i, j, k+1)){
                    if(k + 1 >= size){
                        size = k + 1;
                        idx1 = i;
                        idx2 = j;
                    }
                }
            }
        }
    }

    if(size)
        printf("%d\n(%d,%d)", size, idx1, idx2);
    else
        printf("0\n(-1,-1)");

    return 0;
}
```