# Sitanna

2395

#include <iostream>

using namespace std;

const int maxn = 13, maxm = 1035, INF = 0x7fffffff / 3, bai = 102501, shi = 1024;

const int dirx[] = {0, 0, 1, -1}, diry[] = {1, -1, 0, 0};

int n, m, k, M, map[maxn][maxn], f[maxn][maxn][maxm], fa[maxn][maxn][maxm], ord[maxn][maxn];

bool used[1200130], ans[maxn][maxn];

queue<int>Q;

int calc(int x, int y, int s) {return x \* bai + y \* shi + s;}

bool isSubet(int a, int b) {return (a | b) == a;}

void spfa(int sta) {

while(!Q.empty()) {

int u = Q.front(); Q.pop();

int x = u / bai, y = u % bai / shi;

for(int d = 0; d < 4; d++) {

int nx = x + dirx[d], ny = y + diry[d];

if(nx < 1 || ny < 1 || nx > n || ny > m) continue;

int ns = sta | ord[nx][ny];

if(f[nx][ny][ns] > f[x][y][sta] + map[nx][ny]) {

f[nx][ny][ns] = f[x][y][sta] + map[nx][ny];

fa[nx][ny][ns] = u;

int k = calc(nx, ny, ns);

if(!used[k] && sta == ns) {used[k] = true; Q.push(k);}

}

}

used[u] = false;

}

}

void initForOut(int x, int y, int sta) {

ans[x][y] = true;

int k = fa[x][y][sta];

if(!k) return;

int i = k / bai, j = k % bai / shi, s = k % bai % shi;

initForOut(i, j, s);

if(i == x && j == y) initForOut(i, j, sta - s);

}

void out() {

for(int i = 1; i <= n; i++) {

for(int j = 1; j <= m; j++)

if(ans[i][j])

if(map[i][j]) putchar('o');

else putchar('x');

else putchar('\_'); putchar('\n');

}

}

int main() {

scanf("%d%d", &n, &m);

for(int i = 1; i <= n; i++)

for(int j = 1; j <= m; j++)

for(int s = 0; s < 1024; s++) f[i][j][s] = INF;

for(int i = 1; i <= n; i++)

for(int j = 1; j <= m; j++) {

scanf("%d", &map[i][j]);

f[i][j][0] = map[i][j];

if(!map[i][j]) f[i][j][1 << (k++)] = 0, ord[i][j] = 1 << (k - 1);

}

M = 1 << k;

for(int sta = 1; sta < M; sta++) {

for(int i = 1; i <= n; i++) {

for(int j = 1; j <= m; j++) {

for(int s = 1; s < sta; s++) {

if(isSubet(sta, s)) {

if(f[i][j][sta] > f[i][j][s] + f[i][j][sta - s] - map[i][j]) {

f[i][j][sta] = f[i][j][s] + f[i][j][sta - s] - map[i][j];

fa[i][j][sta] = calc(i, j, s);

}

}

}

if(f[i][j][sta] != INF) Q.push(calc(i, j, sta)), used[calc(i, j, sta)] = true;

}

}

spfa(sta);

}

for(int i = 1; i <= n; i++)

for(int j = 1; j <= m; j++)

if(!map[i][j]) {

printf("%d\n", f[i][j][M - 1]);

initForOut(i, j, M - 1);

out();

return 0;

}

return 0;}