#include <cstdio>

#include <queue>

using namespace std;

int n, m;

int tx, ty, cx, cy;

char s[60][60], S[60][60];

int dx[8] = {-1, -1, -1, 0, 0, 1, 1, 1};

int dy[8] = {-1, 0, 1, -1, 1, -1, 0, 1};

int prod(int x1, int y1, int x, int y, int x2, int y2)

{

int d = (x1-x)\*(y2-y)-(x2-x)\*(y1-y);

if (d == 0)

return 0;

if (d > 0)

return 1;

return -1;

}

bool intersect(int x1, int y1, int x2, int y2, int x3, int y3, int x4, int y4)

{

int d1 = prod(x1, y1, x3, y3, x4, y4);

int d2 = prod(x4, y4, x3, y3, x2, y2);

return d1\*d2 >= 0;

}

queue<int> Qx, Qy, Ql;

int tarx[8], tary[8], l;

int main()

{

freopen("grove.in", "r", stdin);

freopen("grove.out", "w", stdout);

int ans, d, x, y, p, q, len, l, i, j;

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

for (i = 0; i < n; i++)

scanf("%s", S[i]);

for (i = 0; i < n; i++)

for (j = 0; j < m; j++)

if (S[i][j] == '\*')

{

cx = i;

cy = j;

}

ans = n\*m;

for (tx = 0; tx < n; tx++)

for (ty = 0; ty < n; ty++)

if (S[tx][ty] == 'X')

{

while (!Qx.empty())

{

Qx.pop();

Qy.pop();

Ql.pop();

}

l = 0;

for (i = 0; i < n; i++)

for (j = 0; j < m; j++)

s[i][j] = S[i][j];

for (i = 0; i < 8; i++)

{

x = cx+dx[i];

y = cy+dy[i];

if (x < 0 || x >= n || y < 0 || y >= m || s[x][y] == 'X')

continue;

d = prod(x, y, cx, cy, tx, ty);

if (d < 0)

{

Qx.push(x);

Qy.push(y);

Ql.push(1);

s[x][y] = 'X';

}

if (d > 0)

{

tarx[l] = x;

tary[l++] = y;

}

}

while (!Qx.empty())

{

x = Qx.front(); Qx.pop();

y = Qy.front(); Qy.pop();

len = Ql.front(); Ql.pop();

for (i = 0; i < l; i++)

if (x == tarx[i] && y == tary[i])

{

if (ans > len+1)

ans = len+1;

break;

}

if (i != l)

break;

for (i = 0; i < 8; i++)

{

p = x+dx[i];

q = y+dy[i];

if (p >= 0 && p < n && q >= 0 && q < m && s[p][q] == '.')

if (!intersect(x, y, p, q, cx, cy, tx, ty) ||

(tx-p)\*(tx-p)+(ty-q)\*(ty-q)+(tx-cx)\*(tx-cx)+(ty-cy)\*(ty-cy) <=

(cx-p)\*(cx-p)+(cy-q)\*(cy-q))

{

s[p][q] = 'X';

Qx.push(p);

Qy.push(q);

Ql.push(len+1);

}

}

}

}

if (ans == n\*m)

printf("-1\n");

else

printf("%d\n", ans);

return 0;

}