#include <stdio.h>

#include <stdlib.h>

void print\_grid(int \*\*grid, int rows, int cols) {

for (int i = 0; i < rows; i++) {

for (int j = 0; j < cols; j++) {

if (grid[i][j] == 1) {

printf("X ");

} else {

printf(". ");

}

}

printf("\n");

}

}

int count\_neighbors(int \*\*grid, int rows, int cols, int x, int y) {

int count = 0;

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

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

if (i == 0 && j == 0) continue;

int nx = x + i;

int ny = y + j;

if (nx >= 0 && nx < rows && ny >= 0 && ny < cols) {

if (grid[nx][ny] == 1) {

count++;

}

}

}

}

return count;

}

void update\_grid(int \*\*grid, int rows, int cols) {

int \*\*new\_grid = (int \*\*)malloc(rows \* sizeof(int \*));

for (int i = 0; i < rows; i++) {

new\_grid[i] = (int \*)malloc(cols \* sizeof(int));

}

for (int i = 0; i < rows; i++) {

for (int j = 0; j < cols; j++) {

int neighbors = count\_neighbors(grid, rows, cols, i, j);

if (grid[i][j] == 1) {

if (neighbors < 2 || neighbors > 3) {

new\_grid[i][j] = 0;

} else {

new\_grid[i][j] = 1;

}

} else {

if (neighbors == 3) {

new\_grid[i][j] = 1;

} else {

new\_grid[i][j] = 0;

}

}

}

}

for (int i = 0; i < rows; i++) {

for (int j = 0; j < cols; j++) {

grid[i][j] = new\_grid[i][j];

}

}

for (int i = 0; i < rows; i++) {

free(new\_grid[i]);

}

free(new\_grid);

}

int main() {

int rows, cols;

int x, y;

int virus\_count = 5;

printf("Enter the number of rows: ");

scanf("%d", &rows);

printf("Enter the number of columns: ");

scanf("%d", &cols);

int \*\*grid = (int \*\*)malloc(rows \* sizeof(int \*));

for (int i = 0; i < rows; i++) {

grid[i] = (int \*)malloc(cols \* sizeof(int));

}

for (int i = 0; i < rows; i++) {

for (int j = 0; j < cols; j++) {

grid[i][j] = 0;

}

}

printf("Enter 5 virus starting positions (x y):\n");

for (int i = 0; i < virus\_count; i++) {

printf("Virus %d: ", i + 1);

scanf("%d %d", &x, &y);

if (x >= 0 && x < rows && y >= 0 && y < cols) {

grid[x][y] = 1;

} else {

printf("Invalid position! Try again.\n");

i--;

}

}

printf("\nInitial Grid:\n");

print\_grid(grid, rows, cols);

int generations;

printf("Enter the number of generations to simulate: ");

scanf("%d", &generations);

for (int gen = 0; gen < generations; gen++) {

printf("\nGeneration %d:\n", gen + 1);

update\_grid(grid, rows, cols);

print\_grid(grid, rows, cols);

}

for (int i = 0; i < rows; i++) {

free(grid[i]);

}

free(grid);

return 0;

}