#include "estructuraM.h"

#include <iostream>

using namespace std;

EstructuraM crearEstructuraM(int tamano) {

// Verificar que el tamaño sea impar

if (tamano % 2 == 0) {

tamano++; // Aumentar en 1 si es par para asegurar que sea impar

}

EstructuraM estructura;

estructura.filas = tamano;

estructura.columnas = tamano;

// Asignar memoria dinámica para la matriz

estructura.datos = new int\*[tamano];

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

estructura.datos[i] = new int[tamano];

}

// Rellenar la matriz con valores -1 (espacios en blanco)

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

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

estructura.datos[i][j] = -1;

}

}

// Rellenar el centro de la matriz con números del 1 al tamaño^2 excluyendo el centro

int num = 1;

int centro = tamano / 2;

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

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

if (i != centro || j != centro) {

estructura.datos[i][j] = num++;

}

}

}

return estructura;

}

void liberarEstructuraM(EstructuraM &estructura) {

for (int i = 0; i < estructura.filas; i++) {

delete[] estructura.datos[i];

}

delete[] estructura.datos;

estructura.filas = 0;

estructura.columnas = 0;

}

void rotarAntihorario(EstructuraM &estructura, int rotaciones) {

int tamano = estructura.filas;

int \*\*nuevaMatriz = new int\*[tamano];

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

nuevaMatriz[i] = new int[tamano];

}

for (int r = 0; r < rotaciones; r++) {

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

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

nuevaMatriz[tamano - 1 - j][i] = estructura.datos[i][j];

}

}

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

delete[] estructura.datos[i];

}

delete[] estructura.datos;

estructura.datos = nuevaMatriz;

// Asignar nueva memoria para la matriz rotada

nuevaMatriz = new int\*[tamano];

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

nuevaMatriz[i] = new int[tamano];

}

}

// Liberar memoria de la última matriz asignada

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

delete[] nuevaMatriz[i];

}

delete[] nuevaMatriz;

}

void imprimirEstructuraM(EstructuraM &estructura) {

for (int i = 0; i < estructura.filas; i++) {

for (int j = 0; j < estructura.columnas; j++) {

if (estructura.datos[i][j] == -1) {

cout << " "; // Imprimir espacio en blanco

} else {

cout << estructura.datos[i][j] << " ";

}

}

cout << endl;

}

}