#include <iostream>

#include <vector>

#include "cerradura.h"

#include "estructuraM.h"

using namespace std;

Cerradura generarConfiguracionCerradura(const vector<int>& regla) {

int numEstructuras = (regla.size() / 5) + 1; // Determinar número de estructuras necesarias

Cerradura cerradura = configureCerradura(numEstructuras); // Generar configuración inicial

Cerradura configuracionInicial = cerradura;

Cerradura mejorConfiguracion = buscarConfiguracionValida(configuracionInicial, regla);

if (validarRegla(mejorConfiguracion, regla)) {

return mejorConfiguracion;

} else {

cout << "No se encontró una configuración válida que cumpla con la regla." << endl;

return Cerradura(); // Devolver cerradura vacía si no se encuentra solución

}

}

Cerradura buscarConfiguracionValida(const Cerradura& configuracion, const vector<int>& regla) {

if (validarRegla(configuracion, regla)) {

return configuracion; // Devolver la configuración si es válida

}

Cerradura mejorConfiguracion;

int numEstructuras = configuracion.estructuras.size();

int numRotacionesMaximas = configuracion.estructuras[0].filas; // Asumir matrices cuadradas

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

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

Cerradura configuracionRotada = configuracion;

rotarAntihorario(configuracionRotada.estructuras[i], j);

mejorConfiguracion = buscarConfiguracionValida(configuracionRotada, regla);

if (validarRegla(mejorConfiguracion, regla)) {

return mejorConfiguracion;

}

rotarAntihorario(configuracionRotada.estructuras[i], numRotacionesMaximas - j); // Deshacer rotación

}

}

return configuracion; // No se encontró solución, devolver configuración original

}