**Informe Puntos 3 y 4**

**a) Análisis del problema y consideraciones para la alternativa de solución propuesta.**

**Punto 3:** Configurar cerraduras de la cerradura X con estructuras M variables

El problema consiste en desarrollar un módulo que permita configurar cerraduras X compuestas por múltiples estructuras M de diferentes tamaños. La cantidad de estructuras M y sus dimensiones (filas y columnas) deben ser configurables por el usuario.

Para abordar este problema, se propone la siguiente alternativa de solución:

-Crear una estructura Cerradura que contenga un vector de EstructuraM.

-Implementar una función configureCerradura() que solicite al usuario la cantidad de estructuras M a incluir en la cerradura.

-Para cada estructura M, solicitar al usuario el tamaño (filas y columnas) y crear una instancia de EstructuraM utilizando la función crearEstructuraM(int tamano) del punto 1.

-Agregar cada EstructuraM creada al vector dentro de la estructura Cerradura.

-Devolver la Cerradura configurada con las estructuras M especificadas.

**Consideraciones:**

-Reutilizar la función crearEstructuraM(int tamano) del punto 1 para crear cada estructura M dentro de la cerradura.

-Permitir una configuración flexible de la cerradura, donde el usuario pueda especificar la cantidad y tamaño de las estructuras M.

**Punto 4:** Validar una regla K sobre una configuración dada de la cerradura X

El problema consiste en implementar funciones que permitan validar si una regla K se cumple para una configuración dada de la cerradura X.

Para abordar este problema, se propone la siguiente alternativa de solución:

-Implementar una función validarRegla(const Cerradura& cerradura, const vector<int>& regla) que reciba una cerradura configurada y una regla K como parámetros.

-Dentro de la función, verificar si la regla K tiene un formato válido y si los índices de fila, columna y estructura están dentro de los límites permitidos.

-Iterar sobre cada comparación en la regla K y extraer los índices de fila, columna y estructuras M involucradas, así como el signo de comparación.

-Acceder a los valores de las celdas correspondientes en las estructuras M de la cerradura utilizando los índices extraídos.

-Realizar las comparaciones indicadas por la regla K entre los valores de las celdas y verificar si se cumple la condición.

-Devolver true si todas las comparaciones de la regla K se cumplen, o false en caso contrario.

**Consideraciones:**

-Utilizar las funciones implementadas en los puntos 1 y 2 para acceder y manipular las estructuras M dentro de la cerradura.

-Manejar adecuadamente los casos de error, como reglas K con formato incorrecto o índices fuera de rango.

**b) Esquema donde se describen las tareas definidas en el desarrollo de los algoritmos.**

**Punto 3: Configurar cerradura**

**+-----------------------------------+**

**| configure Cerradura() |**

**+-----------------------------------+**

**|**

**V**

**+--------------------------------------------------+**

**| Solicitar cantidad de estructuras M |**

**+--------------------------------------------------+**

**|**

**V**

**+----------------------------------------+**

**| Crear Cerradura vacía |**

**+----------------------------------------+**

**|**

**V**

**+-----------------------------------+**

**| Repetir para cada |**

**| estructura M |**

**+-----------------------------------+**

**|**

**V**

**+-----------------------------------+**

**| Solicitar tamaño de estructura M |**

**+-----------------------------------+**

**|**

**V**

**+--------------------------------------+**

**| Crear EstructuraM con |**

**|crearEstructuraM(tamano) |**

**+--------------------------------------+**

**|**

**V**

**+-----------------------------------+**

**| Agregar EstructuraM a |**

**| la Cerradura |**

**+-----------------------------------+**

**|**

**V**

**+-----------------------------------+**

**| Devolver Cerradura |**

**+-----------------------------------+**

**Punto 4: Validar regla K**

**+----------------------------------------------+**

**| validarRegla(cerradura, regla) |**

**+----------------------------------------------+**

**|**

**V**

**+---------------------------------------------+**

**| Verificar formato válido regla |**

**+---------------------------------------------+**

**|**

**V**

**+----------------------------------------+**

**| Verificar índices dentro de |**

**| límites |**

**+-----------------------------------------+**

**|**

**V**

**+-----------------------------------+**

**| Repetir para cada |**

**| comparación |**

**+-----------------------------------+**

**|**

**V**

**+----------------------------------------------------+**

**| Extraer índices fila, columna, |**

**| estructuras y signo de comparación |**

**+----------------------------------------------------+**

**|**

**V**

**+-----------------------------------------------------+**

**| Acceder valores celdas correspon- |**

**| dientes en estructuras M |**

**+----------------------------------------------------+**

**|**

**V**

**+---------------------------------------------+**

**| Realizar comparación indicada |**

**| por regla K |**

**+---------------------------------------------+**

**|**

**V**

**+------------------------------------+**

**| Si comparación falla, |**

**| devolver false |**

**+------------------------------------+**

**|**

**V**

**+-----------------------------------------+**

**| Si todas comparaciones OK, |**

**| devolver true |**

**+-----------------------------------------+**

**c) Algoritmos implementados.**

**Punto 3: Configurar cerradura**

Función configureCerradura():

Cerradura cerradura

Solicitar al usuario la cantidad de estructuras M

Para cada estructura M:

Solicitar al usuario el tamaño (filas y columnas)

EstructuraM estructura = crearEstructuraM(tamaño)

Agregar estructura a cerradura

Fin Para

Devolver cerradura

Fin Función

**Punto 4: Validar regla K**

Función validarRegla(cerradura, regla):

Si formato de regla no es válido:

Devolver false

Para cada comparación en la regla:

Extraer índices de fila, columna, estructuras y signo de comparación

Si índices fuera de límites:

Devolver false

Obtener valores de celdas correspondientes en estructuras M

Si comparación no se cumple:

Devolver false

Fin Para

Devolver true

Fin Función

**d) Problemas de desarrollo que afrontó.**

**Punto 3: Configurar cerradura**

Durante el desarrollo de la funcionalidad para configurar cerraduras, se enfrentaron los siguientes problemas:

**1. Manejo de entrada de usuario:** Fue necesario implementar una lógica adecuada para solicitar y validar la entrada del usuario al especificar la cantidad de estructuras M y los tamaños de cada una.

**2. Integración con el punto 1:** La función configureCerradura() debía reutilizar la función crearEstructuraM(int tamano) desarrollada en el punto 1. Esto implicó asegurarse de que las interfaces y los tipos de datos fueran compatibles entre ambos módulos.

**3. Gestión de memoria:** Al agregar múltiples estructuras M a la cerradura, se debió prestar atención a la gestión adecuada de la memoria asignada dinámicamente para cada estructura.

**Punto 4: Validar regla K**

En el desarrollo de la funcionalidad para validar la regla K, se presentaron los siguientes problemas:

**1. Formato de la regla K:** Fue necesario implementar una lógica para verificar si el formato de la regla K proporcionada era válido y cumplía con los requisitos especificados.

**2. Manejo de índices:** Al extraer los índices de fila, columna y estructura desde la regla K, se debió asegurar que estos índices estuvieran dentro de los límites permitidos para cada estructura M dentro de la cerradura.

**3. Acceso a valores de celdas:** Para realizar las comparaciones indicadas por la regla K, se requirió acceder a los valores de las celdas correspondientes en las diferentes estructuras M de la cerradura, lo cual implicó un manejo cuidadoso de los índices y la iteración sobre las estructuras.

**4. Detección de errores:** Se implementaron mecanismos para detectar y manejar adecuadamente los casos de error, como reglas K con formato incorrecto o índices fuera de rango.

**e) Evolución de la solución y consideraciones para tener en cuenta en la implementación.**

**Punto 3: Configurar cerradura**

La solución para configurar cerraduras evolucionó a medida que se analizaban los requisitos y se identificaban las interacciones necesarias con las funcionalidades desarrolladas en los puntos anteriores. Inicialmente, se planteó una solución básica que involucraba la creación de una estructura Cerradura y la adición de estructuras M a esta.

Sin embargo, se identificaron aspectos clave que requerían una atención especial, como la integración con la función crearEstructuraM(int tamano) del punto 1, el manejo de la entrada de usuario y la gestión adecuada de la memoria.

Para abordar estos desafíos, se implementaron las siguientes consideraciones:

**1. Reutilización de código:** Se reutilizó la función crearEstructuraM(int tamano) del punto 1 para crear cada estructura M dentro de la cerradura. Esto permitió aprovechar el trabajo previo y mantener una consistencia en la implementación.

**2. Manejo de entrada de usuario:** Se implementó una lógica para solicitar al usuario la cantidad de estructuras M y el tamaño de cada una. Se incluyeron validaciones para asegurarse de que los tamaños ingresados cumplieran con los requisitos (número de filas impar y igual al número de columnas).

**3. Gestión de memoria:** Se prestó atención a la gestión adecuada de la memoria asignada dinámicamente para cada estructura M dentro de la cerradura. Esto se logró mediante la reutilización de la función liberarEstructuraM(EstructuraM &estructura) desarrollada en el punto 1.

**Punto 4: Validar regla K**

La solución para validar la regla K evolucionó a medida que se analizaban los requisitos y se identificaban los desafíos inherentes al problema. Inicialmente, se planteó una solución básica que involucraba la comparación de los valores de las celdas indicadas por la regla K entre las estructuras M correspondientes.

Sin embargo, se identificaron aspectos clave que requerían una atención especial, como la verificación del formato de la regla K, el manejo de índices fuera de rango y la detección de errores.

Para abordar estos desafíos, se implementaron las siguientes consideraciones:

**1. Verificación del formato de la regla K:** Se implementó una lógica para verificar si el formato de la regla K proporcionada era válido y cumplía con los requisitos especificados. Esto incluyó la validación del número de elementos en la regla y el rango de valores permitidos.

**2. Manejo de índices:** Se implementaron mecanismos para verificar que los índices de fila, columna y estructura extraídos de la regla K estuvieran dentro de los límites permitidos para cada estructura M dentro de la cerradura. Esto ayudó a evitar accesos fuera de rango y comportamientos indefinidos.

**3. Acceso a valores de celdas:** Se desarrolló una lógica para acceder a los valores de las celdas correspondientes en las diferentes estructuras M de la cerradura, utilizando los índices extraídos de la regla K. Esto implicó un manejo cuidadoso de los índices y la iteración sobre las estructuras.

**4. Detección de errores:** Se implementaron mecanismos para detectar y manejar adecuadamente los casos de error, como reglas K con formato incorrecto o índices fuera de rango. Esto incluyó la devolución de valores adecuados y la generación de mensajes de error descriptivos.

En resumen, la evolución de la solución para los puntos 3 y 4 se centró en abordar los desafíos específicos identificados durante el desarrollo, como la integración con el código existente, el manejo de la entrada de usuario, la gestión de memoria, la verificación del formato de la regla K, el manejo de índices fuera de rango y la detección de errores. Estas consideraciones permitieron obtener una solución robusta y funcional que cumple con los requisitos establecidos.