

Informe Desafio1

Juan Pablo Betancur Zuluaga

Juan David Morales Galeano

1. Análisis del problema:

A. Encontramos una fórmula la cual nos dice la posición de una matriz (fila,columna) respecto a otra. Es decir, si tenemos una matriz 5×5 (4,3) y una 7×7 , sabremos gracias a la fórmula que la posición en la 7×7 sería (5,4). La fórmula es: $(\text{matrizB} - \text{matrizA})/2$, el resultado es lo que debemos sumar a la posición inicial.

B. Se debe considerar que, si por ejemplo tenemos 2 matrices y buscamos en x posición $1 > 2$, puede llegar el caso en que por más que aumentemos el tamaño y rotemos a matriz 2 no vamos a encontrar que en la posición $x \ 1 > 2$, en este caso debemos de crear un límite, donde si matriz 2 llega a éste límite debemos de aumentar el tamaño de la matriz 1, y volver a matriz 2 a la matriz más pequeña posible para volver a comprar.

C. Para decidir cuál será el tamaño de la primera matriz lo que hicimos fue verificar cuál número era mayor entre la posición ingresada (fila, columna) y luego verificar que ésta sea impar, si no lo es simplemente sumamos 1, y éste será el tamaño para la primera matriz.

2. Tareas que se definieron:

A. Implementar una función que cree las matrices.

B. Implementar una función para girar matrices.

C. Implementar el programa principal.

D. Implementar una función que compruebe el funcionamiento de clave con la cerradura.

3. Algoritmos implementados

A. La función `crearMatriz(int k)` se encarga de crear una matriz cuadrada de tamaño $k \times k$, donde el elemento en el centro de la matriz se inicializa como 0, y los demás elementos se llenan con valores ascendentes a partir de 1.

B. la función `rotarMatriz(unsigned int **matriz, int longitud)` realiza la rotación de una matriz cuadrada de tamaño $\text{longitud} \times \text{longitud}$. La rotación se realiza intercambiando los valores de las filas y columnas de la matriz.

C. la función `ubicacion(int principal, int secundario)` calcula la diferencia entre dos números, utilizando la fórmula $(\text{matrizB} - \text{matrix})/2$.

D. la función `comprobar(int *clave, int *candado, int longitud)` verifica si una clave ingresada coincide con un candado en un escenario específico. Utiliza las funciones anteriores para crear y rotar matrices, y luego compara los elementos de las matrices según una lógica dada por los elementos de la clave y el candado.

4. Problemas de desarrollo

A. La fórmula para encontrar la posición de una `matrizA` respecto a una `matrizB` funciona, pero lo que nos presentaba problemas era en un ciclo donde buscábamos que la posición de `matrizB` no fuera negativa ni fuera mayor a el tamaño de `matrizB`.

B. En el main que hicimos hacía la primera comparación bien, pero al momento de hacer la segunda, y querer aumentar el tamaño de la primera matriz de la comparación, dañamos la primera comparación.

C. Al momento de ejecutar el main nos decía que estábamos leyendo una posición de memoria inválida.

D. El main crasheaba de repente.

5. Evolución de la solución y consideraciones para la implementación.

a. Debemos de tener en cuenta que, si tenemos 3 matrices, si terminamos la comparación de matriz 1 con matriz 2, sigue comparar matriz 2 con matriz 3, si por más que aumentemos el tamaño de matriz 3 no encontramos que se cumpla la condición debemos de aumentar el tamaño de matriz 2, pero pasa una cosa, si aumentamos el tamaño de matriz 2 puede llegar a pasar que en la comparación de matriz 1 y matriz 2 ya no se cumpla la condición, al aumentar el tamaño de matriz 2. Para solucionar esto debemos de, luego de aumentar el tamaño de matriz 2, volver a comprar matriz 1 y matriz 2, luego comprar matriz 2 y matriz 3.

b. En el punto 1. C explicamos nuestra forma de encontrar la primera matriz, pero si se ingresa la posición (2,2) , nuestro programa crea una matriz 3*3, y en ésta la posición (2,2) queda el el centro (0), por lo cual si nos ingresan esa posición debemos de crear una matriz 5*5.