

Pregunta 1: Versión recursiva

Implementación:

En un comienzo las funciones entregadas por los profesores para este programa se encargarán de pedirle al usuario el tamaño del arreglo que desea crear, junto con la distancia que se desea buscar; el arreglo de números al azar será creado por la función entregada **crearArray** y lo guardará en una variable llamada arreglo. Luego dicho arreglo creado será mostrado por pantalla, ahora dar paso a la búsqueda de la distancia.

La búsqueda dentro del arreglo será realizada por la función **distanciaRecursiva**, esta función es una función con estructura de recursión de cola que recibirá como argumentos: el arreglo creado previamente, la distancia a buscar, el tamaño del arreglo además recibirá **i** y **j** que representan los índices para realizar la búsqueda en el arreglo.

El primer paso es establecer el caso base de la recursión por medio de la sentencia **if**, este será cuando la variable **i** y **tamaño** sean iguales, esto significa que estamos revisando un arreglo de un solo elemento, debido a esto no se puede encontrar un par de número que cumplan con la distancia buscada. Por lo anterior, se creará un arreglo de números con capacidad para 3 enteros que almacenará la respuesta de que no se encontró un par de números que cumplan con la distancia, por lo que este arreglo solo contendrá ceros, luego de esto el arreglo es retornado y se sale de la función.

En caso de que no nos encontremos en el caso base podemos encontrarnos en otros dos casos, se explicará el segundo de ellos (último en orden del código) para mejorar la comprensión del código. El segundo caso, es el caso general de la recursión, en este por medio de un **if** se comprueba si el “pivote” en la posición **i** tiene la distancia buscada con el número en la posición **j**, en caso de tener la distancia buscada, se creará un arreglo llamado **respuesta** con capacidad de 3 enteros que almacenará la respuesta final, luego se asigna el valor 1 a la primera posición indicando que se encontró un par de números que cumple con la distancia buscada, después se asignará en la segunda y tercera posición el valor del par de números encontrados, para terminar este arreglo será retornado saliendo de la función. En caso de que los números comparados no tengan la distancia buscada, se hará una llamada recursiva de la función, pero ahora se llama aumentando el valor de **j** en 1 dentro de los argumentos, para que el pivote sea comparado con el siguiente número del arreglo.

El primer caso (**else if**) que se encuentra antes del **else**, sería en caso de que el “pivote” se haya comparado con todos los números del arreglo y no se haya encontrado coincidencia con la distancia, si esto ocurre se volverá a llamar la función **distanciaRecursiva** con los mismo argumentos, pero ahora el índice **i** aumentará en 1 para que el siguiente número del arreglo sea el pivote y se compare con el resto de los números del arreglo. Esta condición debe ser comprobada antes del caso general, ya que en caso de que **j** sea igual a **tamaño**, se intentará acceder a índices fuera del arreglo creado inicialmente.

Luego de que se haya salido de la función **distanciaRecursiva**, el arreglo será almacenado en la variable **resultado** la cual será mostrado por pantalla indicando el resultado de la búsqueda recursiva.

Antes de finalizar se libera la memoria utilizada por ambos arreglos y se retorna cero finalizando el programa.