# https://yt3.ggpht.com/-hYkspDLYXMQ/AAAAAAAAAAI/AAAAAAAAAAA/t8z0vauo_7A/s900-c-k-no/photo.jpghttp://intranet.epn.edu.ec/portal/images/articulos/buo_epn.jpgESCUELA POLITÉCNICA NACIONAL

ESCUELA DE FORMACIÓN DE TECNÓLOGOS

**PROGRAMACION TDSD214**

|  |  |
| --- | --- |
| ASIGNATURA: | Programación |
| PROFESOR: | Ing. Ivonne Maldonado |
| PERÍODO ACADÉMICO: | 2023-B |

# DEBER 3

|  |
| --- |
| TÍTULO:  **ARREGLOS** |
|  |
| Arrays en C++ – I |

**Nombre: Guerra Lovato Josué Eduard**

Tecnología Superior en Desarrollo de Software Deber 3 – Arreglos

## PROPÓSITO DE LA PRÁCTICA

Familiarizar al estudiante con el uso de arreglos en el lenguaje C++.

## OBJETIVO GENERAL

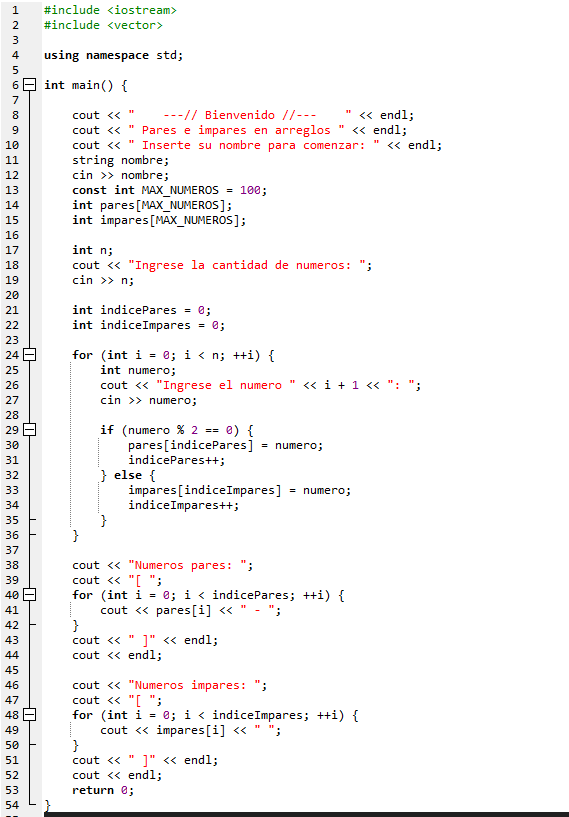
Utilizar arreglos en la solución de problemas, Buscando una comprensión sólida de los conceptos fundamentales relacionados con arreglos, incluyendo, pero no limitado a índices, recorridos, manipulación de elementos y dimensiones.

## OBJETIVO ESPECÍFICOS

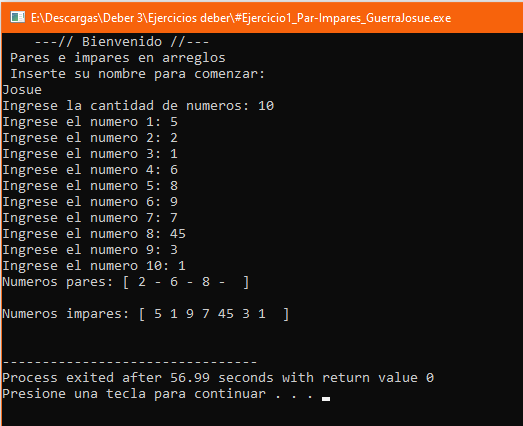
* Manipulación de arreglos multidimensionales.
* Manipulación avanzada de elementos.
* Dominio de índices y recorridos.

## TAREA

* Solicite números al usuario y almacene los pares en un arreglo y los impares en otro arreglo.



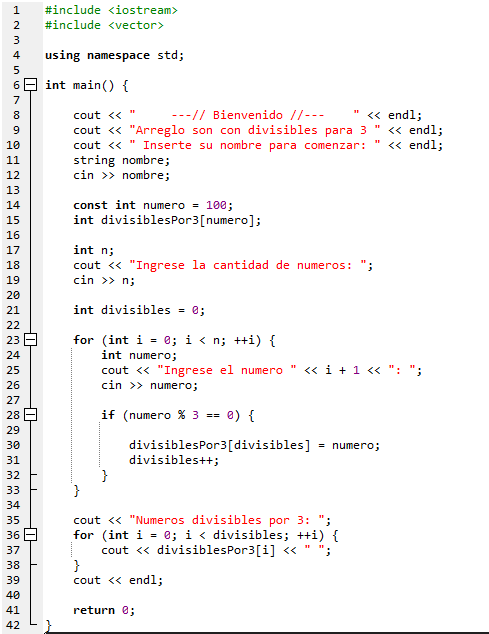
Ejecución:



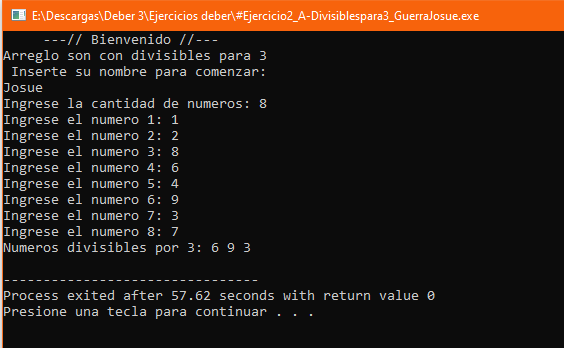
El programa como tal solicita al usuario que inserte la cantidad del arreglo y con eso también los valores del miso de ahí según el número insertado este se retornara como dentro de otro arreglo el cual detecta si es para o impar y así los va ordenando contando desde la posición 0 para acomodar en cada lugar correspondiente, como tal se tiene en cuenta que el código también realiza la operación respectiva para ver si su modulo es o no par, el inicio donde solicita el nombre solo es como parte del programa para darle un poco de formalidad y explicar de manera corta al usuario lo que realizara dentro del programa presentado.

EL EJERCIO 2 SE ENCUENTRA EN LA PÁGINA SIGUIENTE:

* Solicite números al usuario y almacene en un arreglo solo los divisibles para 3.

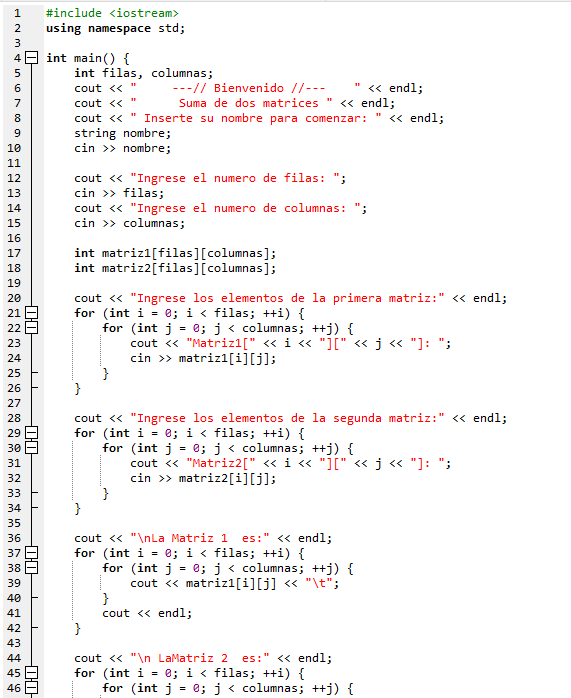


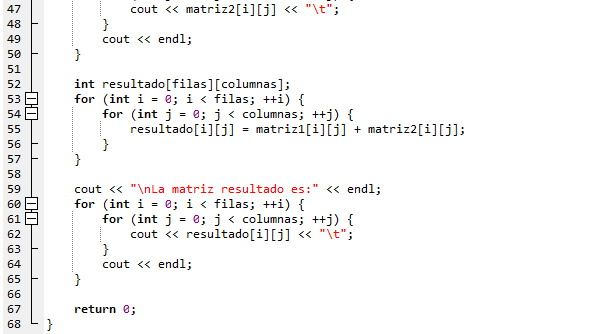
Ejecución:



El programa como tal realizara el calculo correspondiente para saber si el número ingresado es divisible para 3, con eso los ubica dentro de un arreglo y muestra en pantalla dichos números, tomando en cuenta que también se solicita al usuario la cantidad de números que desea ingresar y los mismos cumplen con lo mencionado anteriormente.

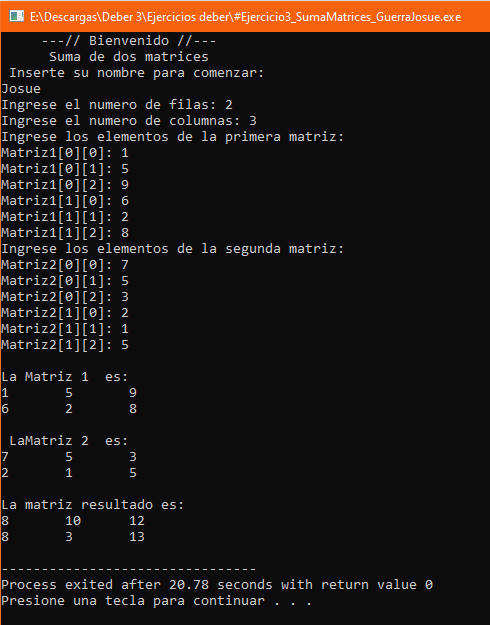
* Sume dos matrices





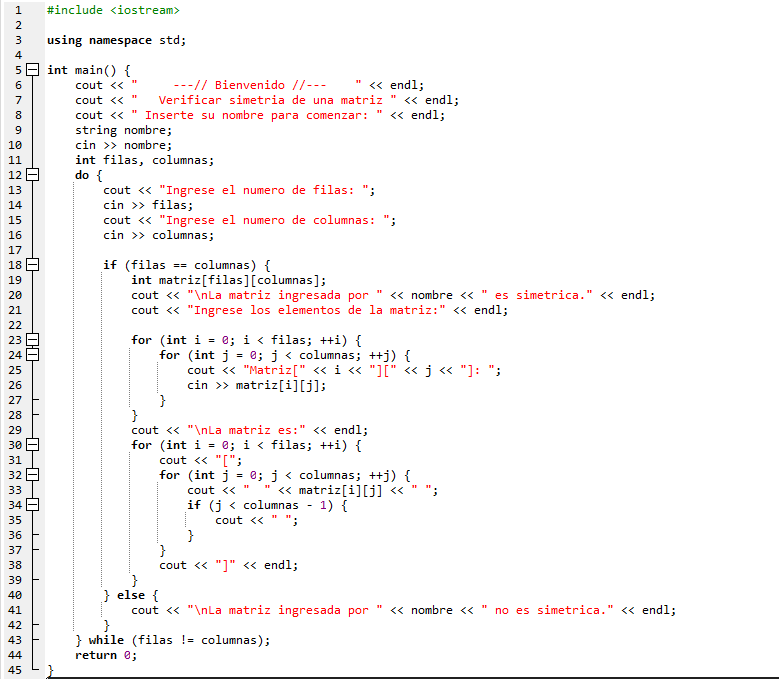
La ejecución se encuentra en la siguiente página

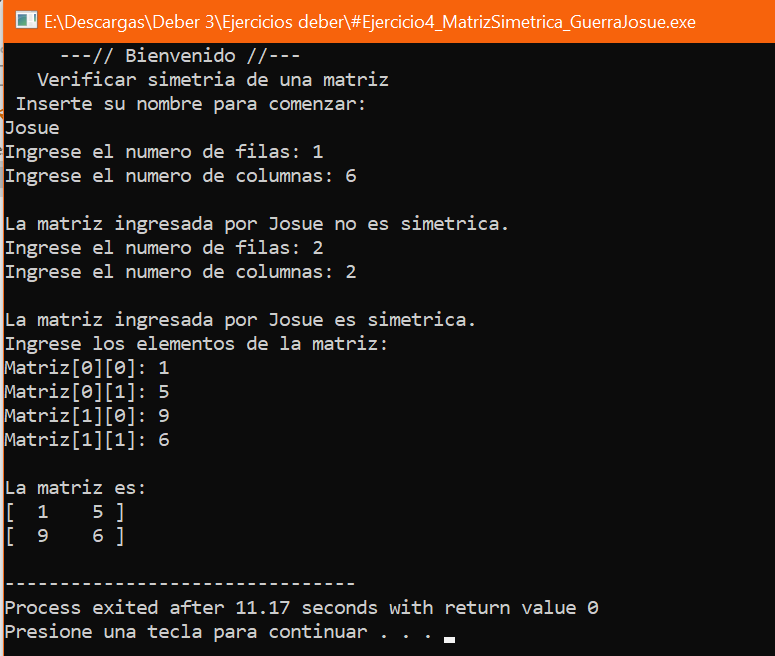
Ejecución:



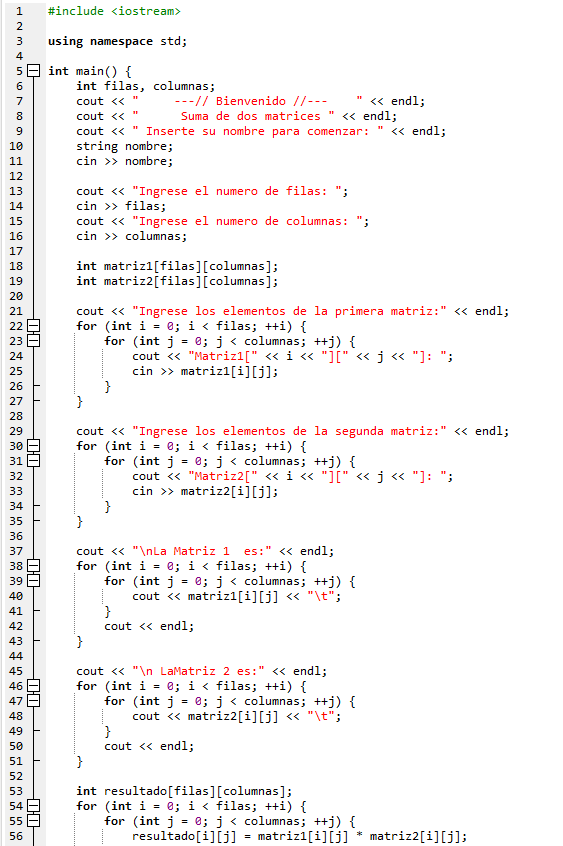
Inicialmente al usuario se le solicita el número de filas y columnas que desea para realizar la operación indicada dentro del problema tomando en cuenta que ambas matrices deben ser de la misma magnitud ya que así se podrán sumar correctamente, y alo referido en magnitud que sean iguales ya que igual pueden tener diferentes números de filas y columnas pero para ambas matrices del mismo tamaño, después de eso se compara y se van sumando los elementos ubicados en la posición de cada matriz y al final mostrara la suma de las mismas.

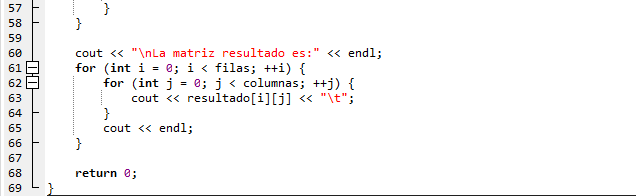
EL EJERCICIO 4 SE ENCUENTRA EN LA SIGUIENTE PÁGINA

* Determine si una matriz es simétrica

Ejecución:

* **Multiplique** dos matrices

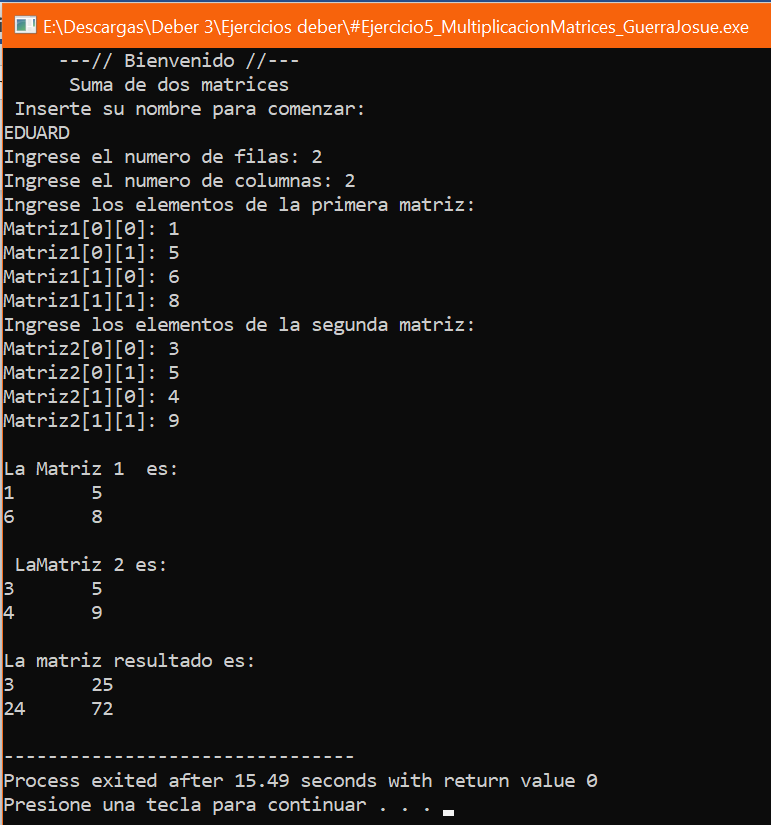




El código toma la misma sintaxis que el ejercicio 3 ya que aplica los mismos parámetros lo único que hace es cambiar el símbolo de la operación y al final muestra la matriz con la operación indicada.

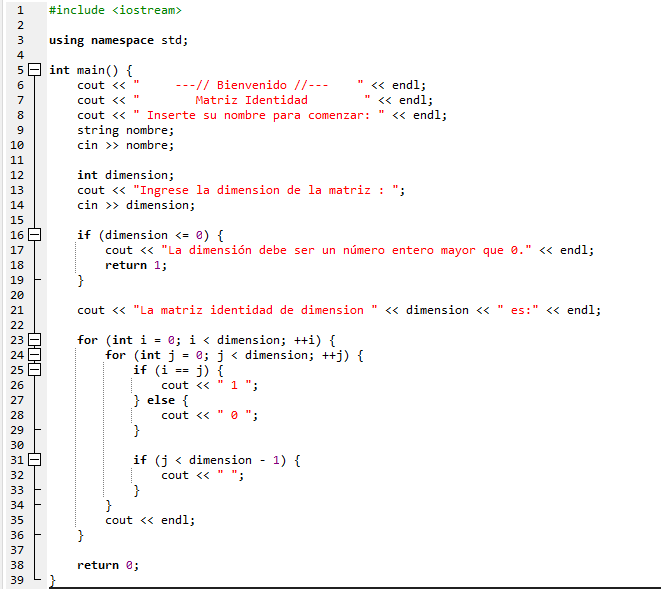
La ejecución se encuentra en la siguiente página.

Ejecución:

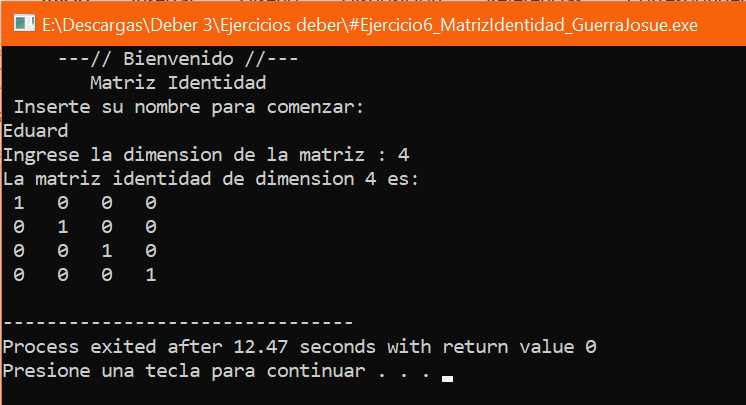


El código toma la misma sintaxis que el ejercicio 3 ya que aplica los mismos parámetros lo único que hace es cambiar el símbolo de la operación y al final muestra la matriz con la operación indicada.

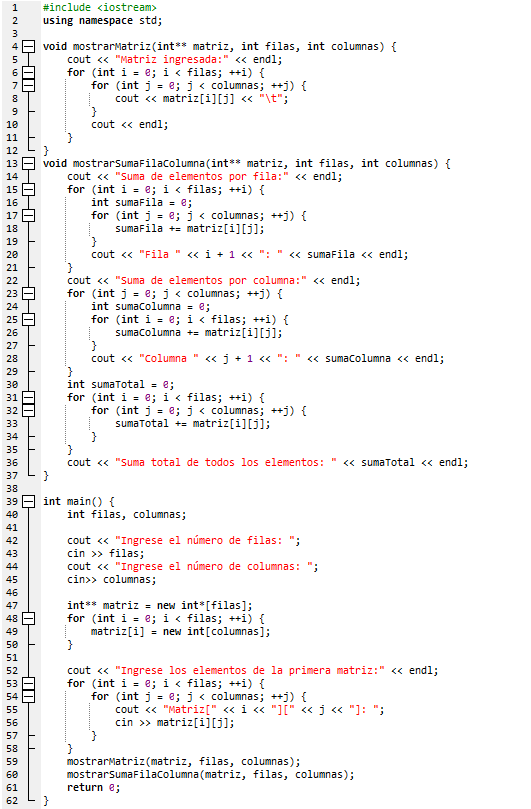
EL EJERCICIO 6 SE ENCUENTRA DESPUÉS DE ESTE SALTO DE PÁGINA.

* Genera e imprime una matriz identidad (una matriz cuadrada con unos en la diagonal principal y ceros en el resto).

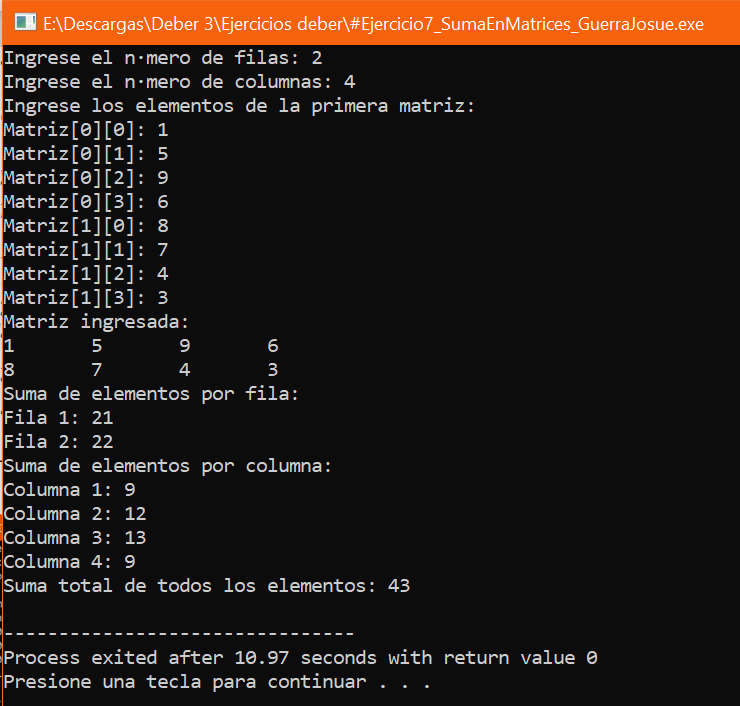
Ejecución:



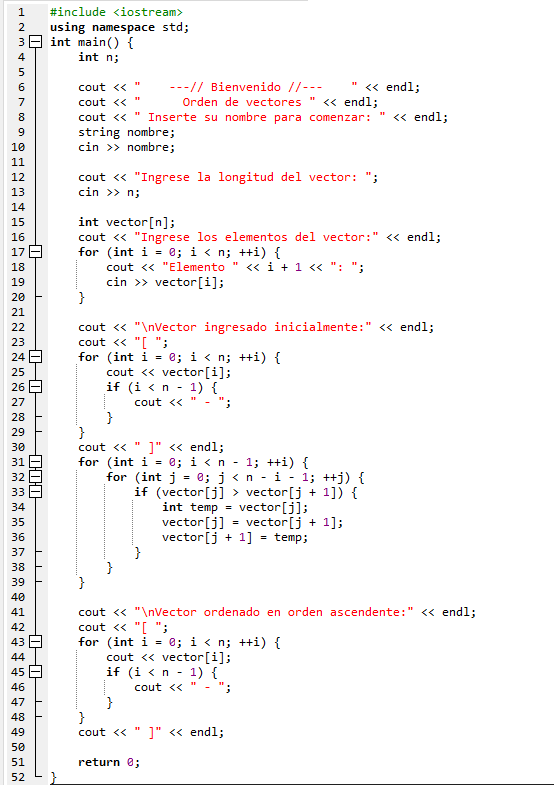
* Suma los elementos de cada fila y cada columna en una matriz y muestre los resultados.



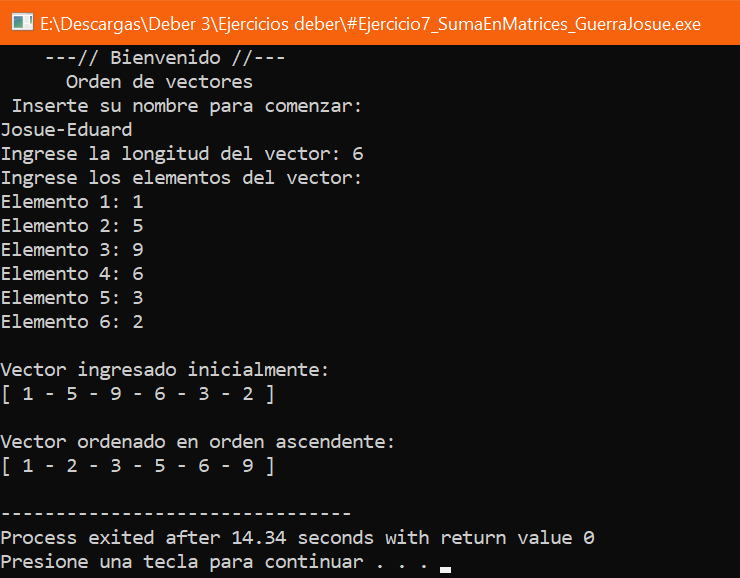
Ejecución:



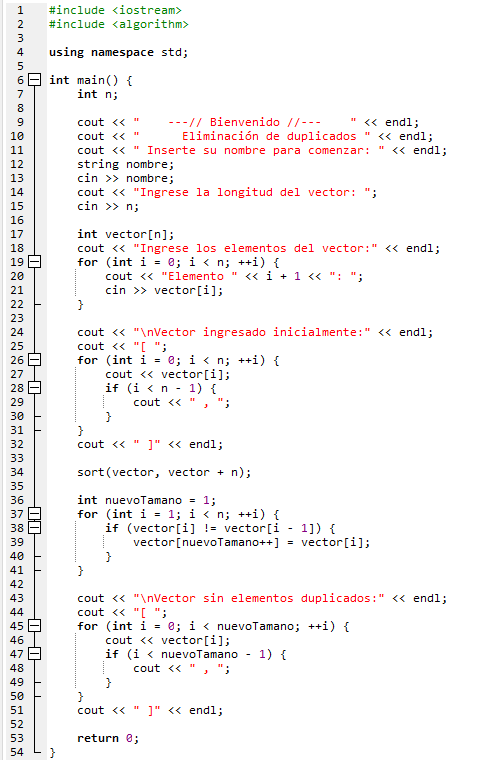
* Ordena de manera ascendente un vector utilizando el algoritmo de ordenamiento burbuja.



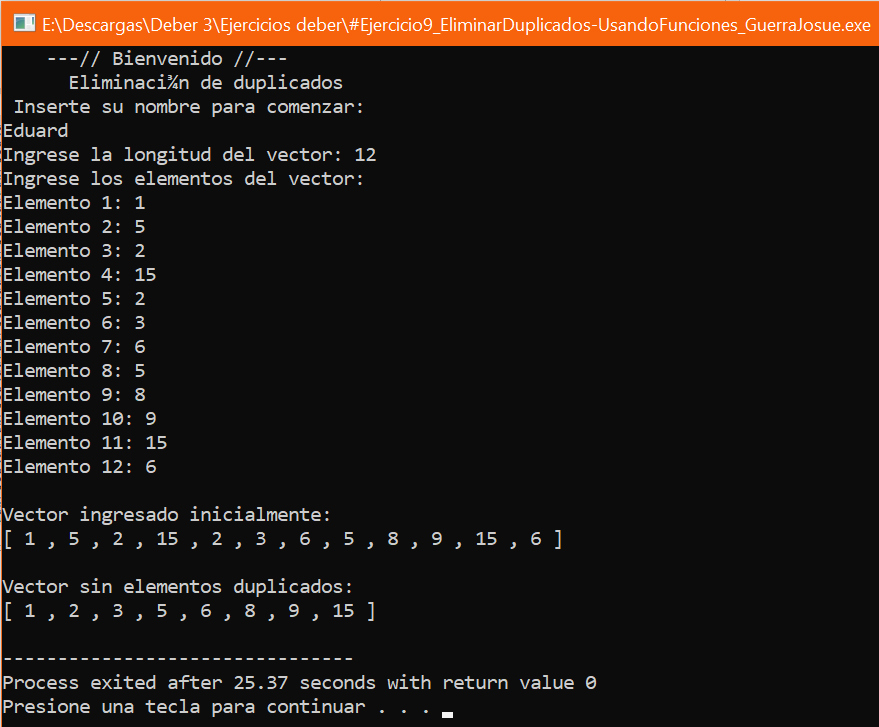
Ejecución:



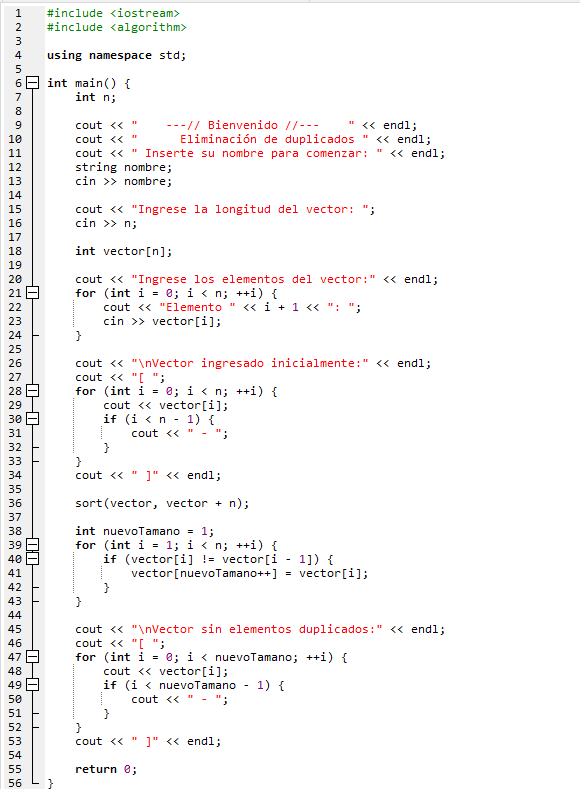
* Elimina los elementos duplicados de un vector y muestre el vector resultante.
* Este ejercicio se realizo de dos maneras diferentes.
* 1 usando funciones para ordenar.



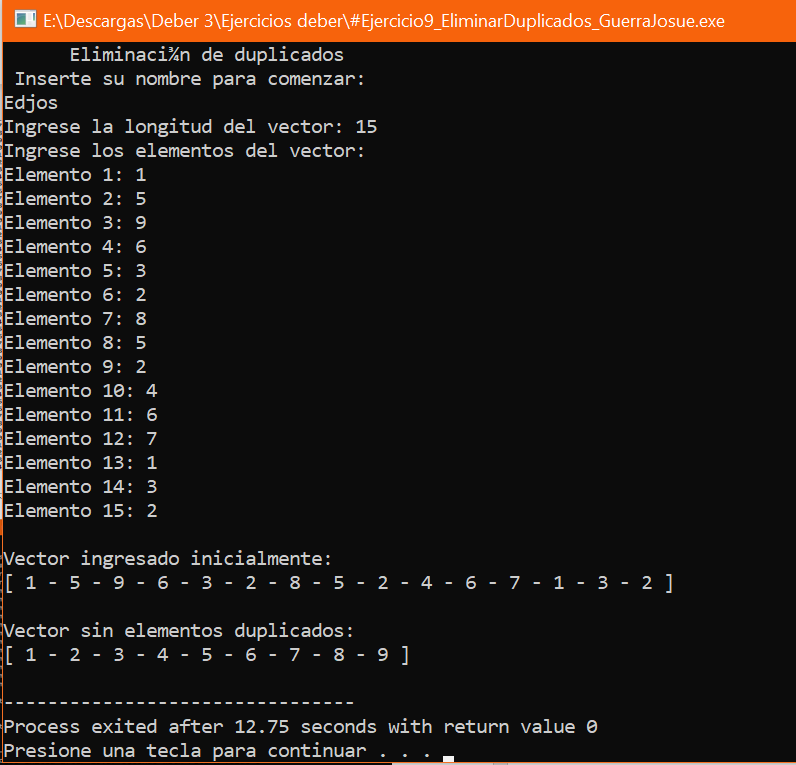
Ejecución:



- 2 Código normal



Ejecución:



## CONSULTA

* Consulte como se retorna un arreglo en una función. (Recuerde esto se debe hacer con punteros ˝◆.)

**RETORNO DE ARREGLOS EN UNA FUNCIÓN**

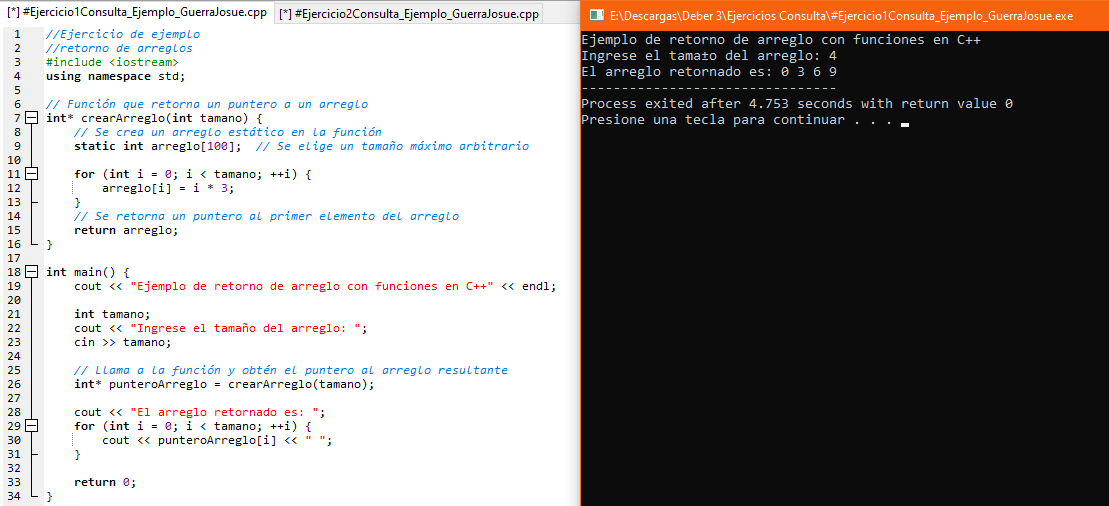
Dentro de c++ no se puede retornar un arreglo directamente de una función, tomando que cuenta que los punteros no pueden llegar a ser copiados del todo, pero se lo puede llegar a devolver mediante un puntero que apunte hacia el primer elemento del arreglo.

En consideración cuando se trabaja con arreglos se trata con punteros, el cual se lo puede llegar a entender como bloque contiguo de memoria como tal siendo una dirección que apunta en el primer arreglo, este procede a apuntar la dirección en memoria.

Las formaciones en c++ son un tipo de datos con un comportamiento particular, se les aplica una regla especial conocida como decaimiento a puntero, que en términos generales viene a decir que si pasas una formación a una función ésta se transformará en puntero, En C/C++, cuando utilizamos la notación array[] como parámetro de una función, es importante comprender que en realidad se trata de un puntero al primer elemento del array proporcionado. En consecuencia, el prototipo de la función debe estar diseñado para devolver un puntero al tipo de dato almacenado en el array, que en este caso es int. Una vez que la función ha completado su ejecución, podemos acceder a los elementos del array mediante la notación de corchetes [] o desreferenciando directamente el puntero resultante.

Este enfoque es esencial para entender cómo se gestionan y manipulan los arrays en funciones, ya que permite trabajar con punteros y acceder a los datos del array de manera eficiente, facilitando así la manipulación y procesamiento de los elementos contenidos en él [1], [2].

* Ponga un ejemplo práctico de retorno de un arreglo.

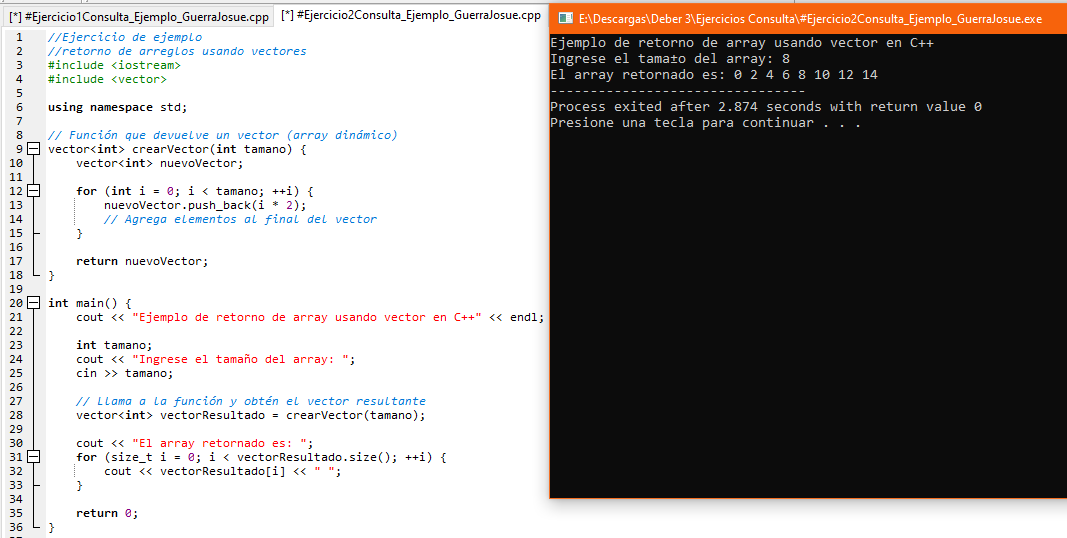


**USAR EL VECTOR PARA DEVOLVER EL ARRAY EN LA FUNCIÓN**

En C++, el contenedor vector es una estructura de datos dinámica que permite almacenar elementos de manera dinámica y flexible. Al utilizar vector para devolver un array desde una función, se evitan problemas asociados con la devolución de punteros a arreglos locales, mitigando así riesgos de comportamiento indefinido. vector proporciona una solución segura y eficiente, ya que gestiona dinámicamente la memoria y se adapta a la cantidad de elementos necesarios, facilitando el manejo de datos de manera robusta y evitando problemas comunes relacionados con la gestión manual de memoria.

Esto simplifica la gestión de memoria, mejora la seguridad y ofrece funcionalidades adicionales, como la capacidad de conocer el tamaño del vector dinámicamente. Al emplear vector, se promueve un código más moderno, mantenible y seguro, permitiendo a los desarrolladores concentrarse en la lógica del problema en lugar de preocuparse por detalles de gestión de memoria. Es crucial explorar y practicar activamente el uso de vector para aprovechar sus ventajas en términos de eficiencia y seguridad en la programación en C++ [2].

* Ponga un ejemplo práctico de retorno de un arreglo.



## PRESENTACIÓN

Al finalizar tu laboratorio deberás subir:

* + Un archivo en formato pdf con el nombre (Deber3\_Programación\_NApellido)

**RECOMENDACIONES**

* Es preferible tener una práctica activa dentro del uso de arreglos en lo que se puede abordar problemas prácticos, participación en proyectos, fortaleciendo comprensión conceptual.
* La colaboración con compañeros de clase y la participación en sesiones de revisión de código pueden ser valiosas. Al analizar y discutir soluciones implementadas por otros, se pueden ganar perspectivas adicionales sobre enfoques y técnicas relacionadas con arreglos.

## CONCLUSIONES

Utilizar arreglos en la solución de problemas, Buscando una comprensión sólida de los conceptos fundamentales relacionados con arreglos, incluyendo, pero no limitado a índices, recorridos, manipulación de elementos y dimensiones.

# Bibliografía

|  |  |
| --- | --- |
| [1] | StackOverflow, «StackOverflow,» 27 04 2020. [En línea]. Available: https://es.stackoverflow.com/questions/349765/c%C3%B3mo-pasar-arrays-a-una-funci%C3%B3n-y-c%C3%B3mo-retornarlos. [Último acceso: 26 01 2024]. |
| [2] | J. Hu, «DelftStack,» 12 10 2023. [En línea]. Available: https://www.delftstack.com/es/howto/cpp/how-to-return-array-from-a-function-in-cpp/#:~:text=C%C3%B3mo%20devolver%20un%20array%20desde%20una%20funci%C3%B3n%20en,devolver%20el%20array%20desde%20la%20funci%C3%B3n%20en%20C%2B%2B. [Último acceso: 26 01 2023]. |

Prof. Ing. Ivonne Maldonado 2