

**TRABAJO PRACTICO N° 2** - Resuelva los ejercicios en Dev C++

Parte A Bucles: Conjunto de instrucciones que se repiten al cumplirse una condición.

Parte A: ARREGLOS - son un tipo de dato que almacena una secuencia de datos del mismo tipo. Sus elementos se almacenan en zonas contiguas de memoria y se puede acceder a ellos de manera directa mediante un índice o posición.

Ejercicio 1: Escribe un programa que con While, defina un vector de números y calcule la suma de sus elementos. Use la librería conio.h, y la función getch().

Ejercicio 2: Escribe un programa usando un for, que defina un vector de números y calcule la multiplicación acumulada de sus elementos. Use la librería conio.h, y la función getch().

Ejercicio 3: Escribe un programa que lea de la entrada estándar un vector de números de tamaño 100, pero que cargue por teclado n números, y los muestre en la salida estándar los números del vector con sus índices asociados.

Ejercicio 4: Escribe un programa que defina un vector de números y muestre en la salida estándar el vector en orden inverso—del último al primer elemento.

Ejercicio 5: Desarrolle un programa que lea de la entrada estándar un vector de enteros y determine el mayor elemento del vector.

Ejercicio 6: Escribe un programa que defina un vector de números y calcule si existe algún número en el vector cuyo valor equivale a la suma del resto de números del vector.

Ejercicio 7: Realiza un programa que defina dos vectores de caracteres y después almacene el contenido de ambos vectores en un nuevo vector, situando en primer lugar los elementos del primer vector seguido por los elementos del segundo vector. Muestre el contenido del nuevo vector en la salida estándar.



Ejercicio 8: Hacer un programa que lea 5 números en un arreglo, los copie a otro arreglo

Ejercicio 9: Se tiene el vector A con 100 elementos almacenados. Diseñe un algoritmo que escriba SI, si el vector esta ordenado ascendentemente, o NO si el vector no está ordenado

Ejercicio 10: Dado un arreglo V de tamaño 20 con números enteros. Deje en un vector A todos los números negativos y en un vector B el doble de los positivos.

Parte B: MATRICES: ¿Qué es una Matriz o Tabla? Una matriz es un vector de vectores o un también llamado array bidimensional. La manera de declarar una matriz es C++ es similar a un vector

Ejercicio 1: Hacer un array bidimensional 3x3, ingresar datos en él, mostrarla y luego mostrar la traspuesta. La traspuesta de una matriz se consigue intercambiando filas por columnas y viceversa.

Ejercicio 2: Realiza un programa que defina una matriz de NxN y escriba un ciclo para que muestre la diagonal principal de la matriz. El número N es un valor válido entre 3 a 7.



Ejercicio 3: Hacer una matriz $M1$ de tipo entera de $2 * 2$, llenarla de números y luego copiar todo el contenido de $M1$ hacia otra matriz $M2$, sumándole 1 a cada elemento de $M1$ al copiarlo.

Ejercicio 4: Hacer una matriz preguntando al usuario el número de filas y columnas, llenarla de números aleatorios, copiar el contenido a otra matriz y por último mostrarla en pantalla.

Ejercicio 5: Realice un programa que lea una matriz de 3×3 y cree su matriz traspuesta. La matriz traspuesta es aquella en la que la columna i era la fila i de la matriz original.

1	2	3
4	5	6
7	8	9

 -->

1	4	7
2	5	8
3	6	9

Ejercicio 6: Realice un programa que calcule la suma de dos matrices cuadradas de 3×3 .

Ejercicio 7: Desarrollar un programa que determine si una matriz es simétrica o no. Una matriz es simétrica si es cuadrada y si es igual a su matriz traspuesta.

Ejercicio 8: Realice un programa que calcule el producto de dos matrices cuadradas de 3×3 .

Ejercicio 9: Hacer un programa en C++ que muestre en pantalla un tablero de ajedrez de la siguiente manera:

1. - Los peones con la letra P.
2. - Las torres con la letra T.
3. - Los caballos con la letra C.
4. - Los alfiles con la letra A.
5. - El rey con la letra M.
6. - La reina con letra M.

Ejercicio 10: Hacer un array bidimensional que acepte nueve números, luego, pedir al usuario que ingrese las coordenadas (x,y) para mostrar el elemento guardado en dicha posición.