

ARREGLOS CON C++

● VECTORES

Estructura de Datos

EJERCICIOS DE APRENDIZAJE

Para cada ejercicio propuesto, primero lleva a cabo un análisis detallado del problema con el fin de determinar qué información será requerida como entrada y qué información será generada como salida. Con base en este análisis, desarrolla un algoritmo en C++ que utilice arreglos para su implementación. Asegúrate de identificar claramente los datos de entrada y salida en cada caso.

1. Realizar un programa que rellene un arreglo con 10 valores ingresados por el usuario y los muestre por pantalla.
2. Realizar un programa que lea 10 números reales por teclado, los almacene en un arreglo y muestre por pantalla la suma, resta y multiplicación de todos los números ingresados al arreglo.
3. Desarrollar un algoritmo para multiplicar dos array numéricos en un tercer array.
4. Desarrollar un algoritmo para ingresar “n” notas en un array y luego calcular el número de aprobados y desaprobados.
5. Escribe un programa que defina un arreglo de números y calcule la multiplicación acumulada de sus elementos
6. Escribe un programa que defina un array de números y muestre en la salida estándar el array en orden inverso (del último al primer elemento).
7. Desarrollar un programa que lea de la entrada estándar un array de enteros y determine el mayor elemento del array.
8. Escribe un programa que defina un array de números y calcule si existe algún número en el array cuyo valor equivale a la suma del resto de números del array.

EJERCICIOS DE ARREGLOS CON BUSQUEDAS

1. Escribe un programa que defina un array de números y calcule si existe algún número en el array cuyo valor equivale a la suma del resto de números del array.
2. Escribir un programa en C++ que implemente la búsqueda lineal para determinar la posición de un número específico dentro del arreglo. Si el número no se encuentra en el arreglo, el programa retorna -1.

Descripción:

Entrada: Un arreglo de enteros, por ejemplo {13, 14, 15, 11, 17, 16, 5}, y un número a buscar, como 11.

Salida: La posición del número buscado en el arreglo, o -1 si el número no está presente.

3. Escribir un programa en C++ que implemente la búsqueda binaria para determinar la posición de un número en un arreglo ordenado. Si el número no se encuentra en el arreglo, el programa retorna -1.

Descripción:

Entrada: Un arreglo de enteros, por ejemplo {2, 5, 8, 12, 16, 23}, y un número a buscar, como 1.

Salida: La posición del número buscado en el arreglo, o -1 si el número no está presente.

4. Realizar un programa que rellene un array de tamaño N, con valores ingresados por el usuario. A continuación, se debe buscar un elemento dentro del arreglo. El programa debe indicar la posición donde se encuentra el valor. En caso que el número se encuentre repetido dentro del arreglo se deben imprimir todas las posiciones donde se encuentra ese valor e indicar el número de veces que se repite. Finalmente, en caso que el número a buscar no está adentro del arreglo se debe mostrar un mensaje.

5. Escribe una función de búsqueda lineal que funcione con un arreglo de cadenas (strings). La función debe buscar una cadena específica y retornar true si la encuentra y false si no la encuentra.

Entradas de ejemplo: Arreglo: ["manzana", "banano", "cereza", "durazno"]

Cadena a buscar: "cereza"

EJERCICIOS DE APRENDIZAJE EXTRA

1. Crear dos arreglos que tengan el mismo tamaño y almacenar en uno de ellos nombres de personas como cadenas. En el segundo arreglo se debe almacenar la longitud de cada uno de los nombres. Mostrar por pantalla cada uno de los nombres junto con su longitud.
2. Crear una función que devuelva la diferencia que hay entre el valor más chico de un arreglo y su valor más grande.
3. Crear un vector que contenga 100 notas de 100 supuestos estudiantes, con valores entre 0 y 20 generadas aleatoriamente. Luego, de acuerdo a las notas contenidas, el programa debe indicar cuántos estudiantes son:
 - a) Deficientes 0-5
 - b) Regulares 6-10
 - c) Buenos 11-15
 - d) Excelentes 16-20

4. Realizar un programa con el siguiente menú y le pregunte al usuario que quiere hacer hasta que ingrese la opción salir.

Llenar arreglo A: Arreglo de tamaño N, se debe llenar de manera aleatoria.

Llenar arreglo B: Arreglo de tamaño N, se debe llenar de manera aleatoria.

Llenar arreglo C: con la suma de A y B. La suma se debe realizar elemento a elemento. Ejemplo: $C = A + B$

Llenar arreglo D: con la resta de los arreglos B y A. La resta se debe realizar elemento a elemento. Ejemplo $D = B - A$

Mostrar: Esta opción debe permitir al usuario decidir que arreglo quiere mostrar: arreglo A, B, C, D.

Salir.

Nota: El rango de los números aleatorios para los arreglos será de [-100 a 100]. La longitud para todos los arreglos debe ser la misma, por lo tanto, esa información sólo se solicitará una vez.