

1. ¿Cuál es la diferencia entre un arreglo unidimensional y uno multidimensional en C++?

Un arreglo unidimensional en C++ es una línea de casillas, donde cada casilla contiene un elemento. Puedes acceder a cada casilla con un solo número, que es como el número de la casilla en la línea.

El arreglo multidimensional es una cuadrícula de casillas, donde cada fila y columna contiene elementos. Para acceder a una casilla específica, necesitas dos números: uno para la fila y otro para la columna.

La diferencia radica en la cantidad de números necesarios para acceder a un elemento. Uno para arreglos unidimensionales y dos (o más) para los arreglos multidimensionales.

2. ¿Cómo se declara un arreglo unidimensional en C++ y cuál es su sintaxis básica?

Arreglo para almacenar números enteros y se necesita espacio para 8 números:

Tipo nombre_arreglo[tamaño]; // declaración arreglo.

Int números[8];

3. Explique cómo se inicializan los elementos de un arreglo en C++ y proporcione un ejemplo.

Para inicializar un arreglo debes crearlo, puedes poner los valores que quieras en las llaves después de su declaración, como si se llena las casillas del arreglo con los valores que quieres desde el principio.

Por ejemplo: si se quiere hacer un arreglo de números enteros con los números del 1 al 7, se haría de esta forma

Int números [7] = {1,2,3,4,5,6,7};

4. ¿Cuál es la sintaxis para acceder a un elemento específico en un arreglo en C++?

El arreglo es como una serie de casillas o compartimientos, y cada casilla tiene un número, si se quiere obtener lo que está en una casilla particular, solo se debe indicar el numero de casilla

Sintaxis: nombre del arreglo seguido de corchetes y dentro del corchete se elige el numero de casilla al cual se desea acceder.

```
Int numero= numeros [3];
```

5. ¿Qué es una matriz en C++ y cómo se declara? Proporcione un ejemplo.

Una matriz es una tabla con filar y columnas, es útil para datos bidimensionales donde se puede acceder a cada elemento utilizando dos índices uno para fila y otro para la columna.

```
Tipo_de_dato nombre_matriz[filas][columnas];
```

```
Int matriz[3][3];
```

6. Describa cómo se inicializan los elementos de una matriz en C++ y proporcione un ejemplo.

Para inicializar una matriz se declara valores. Es como llenar una tabla con números o cualquier otro tipo de datos en el momento en el que creas.

Proporcionar los valores directamente después de declarar la matriz. Es como si llenaras la tabla con los valores que deseas desde el principio.

Nombre_matriz[filas][columna] = valor;

```
int matriz[2][3] = {  
  
    {1, 2, 3},  
  
    {4, 5, 6}  
};
```

7. ¿Cómo se calcula la cantidad de elementos en un arreglo o matriz en C++?

Para un arreglo unidimensional: Puedes calcular la cantidad de elementos dividiendo el tamaño total del arreglo (en bytes) por el tamaño de un solo elemento (también en bytes). Esto te dará la cantidad total de elementos en el arreglo

```
int tamano_arreglo = sizeof(arreglo) / sizeof(arreglo[0]);
```

Para una matriz (arreglo bidimensional): El número total de elementos en una matriz es igual al número de filas multiplicado por el número de columnas.

```
int tamano_filas = sizeof(matriz) / sizeof(matriz[0]);
```

```
int tamano_columnas = sizeof(matriz[0]) / sizeof(matriz[0][0]);
```

```
int total_elementos = tamano_filas * tamano_columnas;
```

8. Explique cómo se puede utilizar un bucle "for" para iterar sobre los elementos de un arreglo en C++

- **inicialización:** Aquí defines una variable que servirá como contador. Por lo general, esta variable se inicializa en 0.
- **Condición de terminación:** Esta es una condición que determina cuándo el bucle debe detenerse. Usualmente, la condición es que el contador sea menor que el tamaño total del arreglo.
- **Incremento (o decremento) del contador:** Aquí especificas cómo cambia el contador después de cada iteración del bucle. Usualmente, incrementas el contador en uno (++i).