



Preparación para la entrevista de estructuras de datos, algoritmos , práctica por tema, C++, Java , Python

## Diferencia entre Estructura y Unión

### C<sup>a</sup>

Nivel de dificultad: Fácil • Última actualización: 25 de mayo de 2022



Estructuras en C Es un tipo de dato definido por el usuario disponible en C que permite combinar elementos de datos de diferentes tipos. Las estructuras se utilizan para representar un registro.

Definición de una estructura: Para definir una estructura, se debe usar la sentencia struct. Esta sentencia define un nuevo tipo de dato, con más de un miembro. El formato de la sentencia struct es el siguiente:

Sigue:

```
struct [nombre de la estructura]
{
    definición de miembro;
    definición de miembro;
    ...
    definición de miembro;
};
```

(O)

```
struct [nombre de la estructura]
{
    definición de miembro;
```



```
    definición de miembro;  
    ...  
    definición de miembro;  
}declaración de variable de estructura;
```

Unión en C Es un tipo de dato especial disponible en C que permite almacenar diferentes tipos de datos en la misma ubicación de memoria. Se puede definir una unión con varios miembros, pero solo uno de ellos puede contener un valor en un momento dado. Las uniones proporcionan una forma eficiente de usar la misma ubicación de memoria para múltiples propósitos.

**Definición de una unión:** Para definir una unión, debe usar la sentencia "union" de la misma manera que para definir una estructura. La sentencia "union" define un nuevo tipo de datos con más de un miembro para su programa. El formato de la sentencia "union" es el siguiente:

```
sindicato [nombre del sindicato]  
{  
    definición de miembro;  
    definición de miembro;  
    ...  
    definición de miembro;  
};
```

(O)

```
sindicato [nombre del sindicato]  
{  
    definición de miembro;  
    definición de miembro;  
    ...  
    definición de miembro;  
}declaración de variable de unión;
```



## Similitudes entre Estructura y Unión

1. Ambos son tipos de datos definidos por el usuario que se utilizan para almacenar datos de diferentes tipos como una sola unidad.
2. Sus miembros pueden ser objetos de cualquier tipo, incluyendo otras estructuras, uniones o matrices. Un miembro también puede consistir en un campo de bits.
3. Tanto las estructuras como las uniones solo admiten la asignación = y sizeof operadores. Las dos estructuras o uniones en la asignación deben tener los mismos miembros y tipos de miembros.
4. Una estructura o una unión se puede pasar por valor a funciones y Devuelto por valor por funciones. El argumento debe ser del mismo tipo que el parámetro de la función. Una estructura o unión se pasa por valor, al igual que una variable escalar, como parámetro correspondiente.
5. Operador "." u operador de selección, que tiene uno de los valores más altos. precedencias, se utiliza para acceder a las variables miembro dentro de ambos tipos de datos definidos por el usuario.

Las diferencias entre Estructura y Unión se muestran a continuación en forma de tabla.  
formato como se muestra a continuación:

	STRUCTURE	UNION
<b>Keyword</b>	The keyword <b>struct</b> is used to define a structure	The keyword <b>union</b> is used to define a union.
<b>Size</b>	When a variable is associated with a structure, the compiler allocates the memory for each member. The size of structure is <b>greater than or equal to the sum of sizes of its members.</b>	when a variable is associated with a union, the compiler allocates the memory by considering the size of the largest memory. So, size of union is equal to the size of largest member.
<b>Memory</b>	Each member within a structure is assigned unique storage area of location.	Memory allocated is shared by individual members of union.
<b>Value Altering</b>	Altering the value of a member will not affect other members of the structure.	Altering the value of any of the member will alter other member values.
<b>Accessing members</b>	Individual member can be accessed at a time.	Only one member can be accessed at a time.
<b>Initialization of Members</b>	Several members of a structure can initialize at once.	Only the first member of a union can be initialized.

```
// Programa en C para ilustrar las diferencias // entre  
estructura y Unión
```

```
#include <stdio.h>  
#include <cadena.h>
```

```
// declarando la estructura struct  
struct_example { int entero; float
```

```
decimal; char  
nombre[20]; };
```

```
// declarando la unión
```

```
unión unión_ejemplo { int
```

```
entero; float decimal;  
char nombre[20]; };
```

```
void main() { //
```

```
creando variable para la estructura // e inicializando la  
diferencia de valores // seis struct struct_example
```

```
s={18,38,"geeksforgeeks"};
```

```
// creando variable para la unión // e inicializando  
valores union union_example  
u={18,38,"geeksforgeeks"};
```

```
printf(" datos de estructura:\n entero: %d\n" "decimal: %.2f\n  
nombre: %s\n", s.entero, s.decimal, s.nombre); printf("\n datos de  
unión:\n entero: %d\n" "decimal: %.2f\n nombre: %s\n", u.entero,  
u.decimal, u.nombre);
```



```
// diferencia dos y tres ▲
```



```
printf("\ntamaño de la estructura: %d\n", tamaño de(s));  
printf("tamaño de la unión: %d\n", tamaño de(u));  
  
// diferencia cinco printf("\n  
Accediendo a todos los miembros a la vez:"); s.integer = 183; s.decimal =  
90; strcpy(s.name,  
"geeksforgeeks");  
  
printf("estructura datos:\n entero: %d\n " "decimal: %.2f\n nombre:  
%s\n", s.entero, s.decimal, s.nombre);  
  
u.entero = 183; u.decimal =  
90; strcpy(u.nombre,  
"geeksforgeeks");  
  
printf("\nunión datos:\n entero: %d\n " "decimal: %.2f\n  
nombre: %s\n", u.entero, u.decimal, u.nombre);  
  
printf("\n Accediendo a un miembro a la vez:");  
  
printf("\ndatos de la estructura:"); s.integer =  
240; printf("\nentero: %d",  
s.integer);  
  
s.decimal = 120;  
printf("\ndecimal: %f", s.decimal);  
  
strcpy(s.nombre, "Programación en C");  
printf("\nnombre: %s\n", s.nombre);  
  
printf("\n datos de la unión:"); u.integer =  
240; printf("\ninteger: %d",  
u.integer);  
  
u.decimal = 120;  
printf("\ndecimal: %f", u.decimal);  
  
strcpy(u.nombre, "Programación en C");  
printf("\nnombre: %s\n", u.nombre);  
  
//diferencia cuatro  
printf("\nAlterando un valor de miembro:\n ▲ ");
```



```
s.integer = 1218;  
printf("estructura datos:\n entero: %d\n " " decimal: %.2f\n  
nombre: %s\n", s.integer, s.decimal, s.nombre);
```

```
u.integer = 1218; printf("unión  
datos:\n entero: %d\n" " decimal: %.2f\n nombre: %s\n",  
u.integer, u.decimal, u.nombre); }
```

Producción:

datos de estructura:

entero: 18

decimal: 38.00

nombre: geeksforgeeks

datos sindicales:

entero: 18

decimal: 0.00

nombre: ?

tamaño de la estructura: 28

tamaño de la unión: 20

Acceder a todos los miembros a la vez: datos de estructura: entero:

183

decimal: 90.00

nombre: geeksforgeeks

datos sindicales:

entero: 1801807207

decimal: 2773228717211595100000000000.00

nombre: geeksforgeeks

Acceder a un miembro a la vez:

datos de estructura:



entero: 240  
decimal: 120.000000  
nombre: Programación en C

datos sindicales:  
entero: 240  
decimal: 120.000000  
nombre: Programación en C

Modificar el valor de un miembro:  
datos de estructura:  
entero: 1218  
decimal: 120.00  
nombre: Programación en C  
datos sindicales:  
entero: 1218  
decimal: 0.00  
nombre: ?

Nota: las estructuras son mejores que las uniones ya que la memoria se comparten  
a unión que resulta un poco más rápida. Pero técnicamente  
hablando, sindicatos Mejor porque ayudan a obtener una para guardar mucho,  
ventaja general sobre las estructuras a largo plazo. correr.

[Estructuras de cuestionario y Unión.](#)

Este artículo es una contribución de Harish Kumar. Si te gusta GeeksforGeeks  
y te gustaría contribuir, también puedes escribir un artículo usando  
[write.geeksforgeeks.org](https://write.geeksforgeeks.org) o envía tu artículo a [review-team@geeksforgeeks.org](mailto:review-team@geeksforgeeks.org). Verás tu  
artículo publicado en



Página principal de GeeksforGeeks y ayuda a otros Geeks.


Por favor, escribe comentarios si encuentras algo incorrecto o si quieres...

compartir más información sobre el tema tratado anteriormente.

```
If (Coding)
{
  C foundation course = true;
  Focus = 100;
}
cout << "Success" ;
```

Wait no more!

Start Learning



Me gusta 87

Próximo

Unión en C

## Compatibilidad del horóscopo

¿Eres compatible con tu pareja y  
¿Otras personas? Haz el test - Descúbrelo  
¡la respuesta!

Nebulosa

[Más información](#)

## ARTÍCULOS RECOMENDADOS

Página: 1 2 3



01 Unión Anónima y  
Estructura en C

05

C | Estructura y Unión | Pregunta 4



29 de enero de 2016

04, 13 de febrero

02

C | Estructura y Unión | Pregunta 1

04, 13 de febrero

06

C | Estructura y Unión | Pregunta 5

09, 13 de febrero

03

C | Estructura y Unión | Pregunta 2

04, 13 de febrero

07

C | Estructura y Unión | Pregunta 6

07, Mar 13

04

C | Estructura y Unión | Pregunta 10

04, 13 de febrero

08

C | Estructura y Unión | Pregunta 7

14 de marzo de 2013



Artículo aportado por:



Geeks para Geeks

Votar por la dificultad

Dificultad actual: **Fácil**



GeeksforGeeks

A-143, Piso 9, Torre Corporativa Sovereign,  
Sector 136, Noida, Uttar Pradesh - 201305

Fácil

Normal

Medio

Duro

Comentarios de expertos@geeksforgeeks.org

Mejorado por: [arvindchoudhary6572](#), [qeerransoo](#), [anikaseth98](#), [simmytarika5](#), [21eg110a25](#)

### Compañía

Etiquetas del artículo: [Lenguaje C](#), [Diferencia entre](#)

[Sobre nosotros](#)

[Carreras](#)

[Mejorar el artículo](#)

[Enlace de medios](#)

[Informar de un problema](#)

[Contáctenos](#)

[política de privacidad](#)

[Política de derechos de autor](#)

### Aprender

[Algoritmos](#)

[Estructuras de datos](#)

[Hoja de trucos de SDE](#)

[aprendizaje automático](#)

[Asignaturas de informática](#)

[Tutoriales en video](#)

[Cursos](#)

¿Escribes código en un comentario? Usa [ide.geeksforgeeks.org](https://ide.geeksforgeeks.org), [generar enlace](#) y compartir el enlace aquí.

### Noticias

[Noticias principales](#)

[Tecnología](#)

[Trabajo y carrera](#)

[Negocio](#)

[Finanzas](#)

[Estilo de vida](#)

[Conocimiento](#)

### Idiomas

[Pitón](#)

[Java](#)

[Golang](#)

[DO#](#)

[SQL](#)

[Kotlin](#)



## Desarrollo web

Tutoriales web

Tutorial de Django

HTML

JavaScript

Oreja

ReactJS

NodeJS

## Contribuir

Escribe un artículo

Mejorar un artículo

Elige temas para escribir

Escribe la experiencia de la entrevista

Pasantías

Pasantía de video

@geeksforgeeks , Algunos derechos reservados

