

1. Array de enteros con inicialización parcial

Define un array de 12 números enteros con nombre `num` y asigna valores solo a algunos elementos, por ejemplo:

```
num[0] = 39; num[1] = -2; num[4] = 0; etc.
```

Muestra el contenido completo del array.

Pregunta: ¿Qué sucede con los elementos que no fueron inicializados explícitamente?

En C, si se declara un array como variable global, sus elementos no inicializados valen 0. Pero si se declara como variable local (dentro de `main()`), los elementos **no tendrán un valor predecible** (pueden contener basura de memoria).

2. Array de caracteres parcialmente inicializado

Define un array de 10 caracteres llamado `simbolo` y asigna valores solo a ciertos elementos.

Ejemplo:

```
simbolo[0] = 'a'; simbolo[1] = 'x'; simbolo[4] = '@'; etc.
```

Muestra el contenido de todos los elementos.

Pregunta: ¿Qué contienen los elementos no inicializados?

Al igual que en el caso anterior, si el array se declara localmente, los elementos no inicializados pueden tener valores basura.

3. Mostrar números en orden inverso

Escribe un programa que lea 10 números enteros por teclado y luego los muestre en orden inverso.

Ejemplo de entrada: 5 8 1 4 7 9 3 6 2 0

Salida esperada: 0 2 6 3 9 7 4 1 8 5

4. Arrays de números, cuadrados y cubos

Define tres arrays de 20 enteros: `numero`, `cuadrado` y `cubo`.

- Carga `numero` con valores aleatorios entre 0 y 100.
- Llena `cuadrado` con el cuadrado de cada valor de `numero`.
- Llena `cubo` con el cubo de cada valor de `numero`.

Muestra el contenido de los tres arrays en columnas.

Para generar números aleatorios en C:

```
#include <stdlib.h>

#include <time.h>

srand(time(NULL)); // Inicializar semilla

int r = rand() % 101; // Número entre 0 y 100
```

5. Mostrar máximo y mínimo

Escribe un programa que pida 10 números al usuario.
Luego, muestra los números ingresados indicando cuál es el **máximo** y cuál el **mínimo** con las palabras correspondientes al lado.

Ejemplo:

```
23
45 ← máximo
12
9 ← mínimo
...
```

6. Rotar array una posición a la derecha

Escribe un programa que lea 15 números, los almacene en un array y luego **rote todos los elementos una posición a la derecha**.
El último elemento debe pasar a la primera posición.
Muestra el array resultante.

Rotar a la derecha significa que cada valor se mueve a la posición siguiente, y el último se coloca en la primera posición. Usar un bucle y una función.

Ejemplo:

```
Antes: [1, 2, 3, 4]
Después: [4, 1, 2, 3]
```

7. Mostrar par o impar

Escribe un programa que pida 8 números enteros y muestre cada uno con la palabra **"par"** o **"impar"** al lado, según corresponda.

8. Separar pares e impares

Escribe un programa que genere 20 números enteros aleatorios entre 0 y 100 y los almacene en un array.
Luego, coloca **todos los pares al principio del array** y **todos los impares al final**.
Puedes usar arrays auxiliares si lo necesitas.

Pistas:

- Usa `rand()` para generar los números (ver ejemplo más arriba).
- Podés crear dos arrays auxiliares...

9. Rotar array hasta que un número esté en la posición 0

Escribe un programa que genere un array de 10 enteros aleatorios entre 0 y 100.

Luego pide al usuario que ingrese un número.

Si el número **no está en el array**, se vuelve a pedir.

Cuando finalmente se ingresa un número válido, se debe **rotar el array hacia la derecha** tantas veces como sea necesario para que ese número quede en la posición 0.

Muestra el array resultante.

Qué significa rotar: Mover todos los elementos una posición hacia la derecha (último → primero). Se repite hasta que el número buscado esté en la primera posición.

Pistas:

- Usá un bucle para buscar el número ingresado en el array.
- Si está, guardá su posición.
- Realizá rotaciones hasta que dicho número esté en el índice 0.
- Para rotar, podés hacer una función que:
 1. Guarde el último elemento.
 2. Desplace todos los elementos a la derecha.
 3. Ponga el último elemento en `array[0]`.

Ejercicios de Arrays Bidimensionales

Ejercicio 1 – Cargar manualmente y mostrar una matriz

Crea un array bidimensional de 3 filas por 4 columnas de enteros. Asigná manualmente algunos valores a ciertas posiciones (por ejemplo: `[0][0]`, `[1][2]`, etc.), dejando el resto vacíos.

Mostrá el contenido de la matriz con `printf`.

Pista:

- Los elementos que no se inicialicen tendrán valores basura si el array es local.
 - Usá dos bucles `for` anidados para recorrer filas y columnas.
 - Usá `\n` dentro de `printf` para hacer los saltos de línea y que quede en forma de tabla.
-

Ejercicio 2 – Llenar toda la matriz con teclado

Crea un array de 2 filas por 5 columnas. Pedí al usuario que ingrese números enteros y completá toda la matriz con esos valores.
Luego mostrala en formato de tabla.

Pista:

- Dos `for` anidados: uno para filas y otro para columnas.
 - Cuidado con los índices: `[fila][columna]`.
 - Usá `printf("\n")` después del bucle de columnas para separar bien cada fila.
-

Ejercicio 3 – Centro universitario

En un centro universitario hay 3 carreras: **Programación, Alimentos y Seguridad e higiene**. Cada carrera tiene 3 años.
Crea una matriz de 3x3 que almacene la cantidad de alumnos por año y carrera.
Luego mostrala como tabla y calculá:

- La cantidad total de alumnos por carrera (suma por fila).
- La cantidad total por año (suma por columna).
- La carrera con **mayor cantidad de alumnos**.
- El total general de estudiantes.

Pista:

- Usá índices `matriz[fila][columna]` donde:
 - Filas = carreras
 - Columnas = años
 - Guardá totales en variables auxiliares.
 - Usá `printf("\n")` para que se vea como una tabla.
-

Ejercicio 4 – Cargar aleatoriamente y hallar mínimo y máximo

Crea una matriz de 6 filas por 10 columnas con números aleatorios entre **0 y 1000**.
Mostrá la matriz y luego informá:

- El valor máximo y su posición (fila y columna).
- El valor mínimo y su posición.

Pistas:

- Usá `rand() % 1001` para obtener números de 0 a 1000.
- Inicializá `srand(time(NULL))` al principio.

- Usá variables `max`, `min`, `filaMax`, `colMax`, etc. para guardar los valores y sus posiciones.
 - Dos bucles `for` anidados para recorrer.
-

Ejercicio 5 – Diagonal principal de una matriz 10x10

Crea una matriz de 10x10 con números aleatorios entre 200 y 300.
Mostrá la matriz. Luego:

- Mostrá los elementos que están en la **diagonal principal** (de `[0][0]` a `[9][9]`).
- Calculá el **máximo**, **mínimo** y **promedio** de los elementos de esa diagonal.

Pistas:

- La diagonal principal cumple que `fila == columna`.
- Acumulá los valores en una variable `suma` para calcular el promedio.