En el lenguaje de programación C, **scanf** es una función de la biblioteca estándar (stdio.h) que se utiliza para **leer datos de entrada desde el teclado** (o cualquier flujo de entrada estándar) y almacenarlos en variables. Es la contraparte de printf, que se usa para imprimir datos en la salida estándar.

**Sintaxis de scanf:**

c

Copy

int scanf(const char \*formato, dirección\_de\_la\_variable);

* **formato**: Es una cadena de caracteres que especifica el tipo de dato que se va a leer. Contiene **especificadores de formato** que indican cómo se deben interpretar los datos de entrada.
* **dirección\_de\_la\_variable**: Es la dirección de memoria de la variable donde se almacenará el valor leído. Se usa el operador & para obtener la dirección de la variable.

**Especificadores de formato comunes en scanf:**

| **Especificador** | **Tipo de dato** |
| --- | --- |
| %d | Entero (int) |
| %f | Punto flotante (float) |
| %lf | Punto flotante (double) |
| %c | Carácter (char) |
| %s | Cadena de caracteres |
| %u | Entero sin signo (unsigned int) |
| %ld | Entero largo (long) |

**Ejemplo básico de uso de scanf:**

c

Copy

#include <stdio.h>

int main() {

int edad;

float altura;

char nombre[50];

// Solicitar datos al usuario

printf("Ingrese su edad: ");

scanf("%d", &edad); // Lee un entero y lo almacena en 'edad'

printf("Ingrese su altura (en metros): ");

scanf("%f", &altura); // Lee un float y lo almacena en 'altura'

printf("Ingrese su nombre: ");

scanf("%s", nombre); // Lee una cadena de caracteres y la almacena en 'nombre'

// Mostrar los datos ingresados

printf("\nDatos ingresados:\n");

printf("Nombre: %s\n", nombre);

printf("Edad: %d\n", edad);

printf("Altura: %.2f metros\n", altura);

return 0;

}

**Explicación del ejemplo:**

1. **scanf("%d", &edad)**:
   * Lee un número entero ingresado por el usuario y lo almacena en la variable edad.
   * El especificador %d indica que se espera un entero.
   * El operador & se usa para pasar la dirección de memoria de la variable edad.
2. **scanf("%f", &altura)**:
   * Lee un número decimal ingresado por el usuario y lo almacena en la variable altura.
   * El especificador %f indica que se espera un valor de tipo float.
3. **scanf("%s", nombre)**:
   * Lee una cadena de caracteres ingresada por el usuario y la almacena en el arreglo nombre.
   * El especificador %s indica que se espera una cadena de caracteres.
   * **Nota**: scanf con %s lee solo una palabra (hasta el primer espacio en blanco).

**Consideraciones importantes:**

1. **Uso del operador &**:
   * Es obligatorio usar & antes del nombre de la variable, excepto cuando se trabaja con cadenas de caracteres (%s), ya que el nombre de un arreglo es en sí mismo un puntero a su primer elemento.
2. **Lectura de cadenas con espacios**:
   * scanf con %s no puede leer cadenas con espacios. Para leer una línea completa (incluyendo espacios), se puede usar fgets:

c

Copy

fgets(nombre, 50, stdin);

1. **Validación de entrada**:
   * scanf retorna el número de elementos leídos correctamente. Si la entrada no coincide con el formato esperado, scanf puede fallar. Es recomendable verificar su valor de retorno:

c

Copy

if (scanf("%d", &edad) != 1) {

printf("Error: Entrada no válida.\n");

}

1. **Buffer de entrada**:
   * scanf puede dejar caracteres no deseados en el buffer de entrada (por ejemplo, saltos de línea). Para limpiar el buffer, se puede usar getchar() o fflush(stdin) (aunque fflush(stdin) no es estándar).

**Ejemplo con validación:**

c

Copy

#include <stdio.h>

int main() {

int edad;

printf("Ingrese su edad: ");

if (scanf("%d", &edad) != 1) {

printf("Error: Debe ingresar un número entero.\n");

} else {

printf("Edad ingresada: %d\n", edad);

}

return 0;

}

En resumen, scanf es una función muy útil en C para leer datos de entrada, pero requiere cuidado al manejar formatos y validar la entrada para evitar errores.