

Facultad de Ingeniería de Sistemas Computacionales

Capítulo I Identificadores y palabras claves





Contenido

- Elementos básicos
- Identificadores
- Palabras claves



Identificadores



"Un identificador es cualquier palabra no reservada que comience por una letra o por un subrayado, pudiendo contener en su interior letras, números y subrayados".

Características de C

- C es un lenguaje sensible al contexto por lo que diferencia entre mayúsculas y minúsculas.
- La longitud máxima de un identificador generalmente es de 32 caracteres.

Ejemplos:

- Nombres de variables
- Nombre de las Funciones

(Enrique Vicente Bonet Esteban, 2018)



Palabras Reservadas



El lenguaje C posee un total de 32 palabras reservadas que define el standard ANSI-C.

4		0 1 11	47	25
1.	auto	9. double	17. struct	25. int
2.	break	10. else	18. switch	26. long
3.	case	11. enum	19. typedef	27. registe
4.	char	12. extern	20. union	28. return
5.	const	13. float	21. unsigned	29. short
6.	continue	14. for	22. void	30. signed
7.	default	15. goto	23. volatile	31. sizeof
8.	do	16. if	24. while	32. static

ANSI C es un estándar publicado por el Instituto Nacional Estadounidense de Estándares



Elementos básicos



Identificadores

- Nombres de variables
- Nombre de las Funciones
- Nombre del programa
- Nombre de constantes

Palabras reservadas

- if
- else
- char
- string
- float
- int
- printf
- False
- True
- function
- for
- while
- scanf
- do
- etc.....

• Caracteres especiales

- Coma
- Apóstrofo
- Comilla
- Punto y coma







Toda variable en C se declara de la forma:

<tipo de dato> <nombre de variable> [, nombre de variable];

Tipo	Descripción
char	Carácter o entero pequeño (byte)
int	Valor numérico entero sin decimal
float	Punto flotante (valor numérico con decimales)
double	Punto flotante (mayor rango que float)
void	Sin tipo (uso especial)

Ejemplos:
float salario;
int edad, cantidad;
char letra;

(Enrique Vicente Bonet Esteban, 2018)

Identificadores y palabras claves



Tipos de datos en C

OF SELHORS COM	
1 Tag	
3	
Q	
34	
NO DESCRIPTION OF THE PARTY.	
TOWN TOWN OF THE PARTY OF THE P	

Tipo	Tamaño en memoria	Valores posibles	Descripción
char	1 byte	-128 a 127 o 0 a 255	Guarda caracteres del codigo ASCII
int	2 or 4 bytes	-32,768 a 32,767 o -2,147,483,648 a 2,147,483,647	Guarda numeros enteros,
float	4 bytes	1.2E-38 a 3.4E+38	Guarda numeros decimales con precisión de 6 digitos decimales
short	2 bytes	-32,768 a 32,767	Guarda numeros enteros con menor capacidad de almacenamiento(y rango) que int
double	8 bytes	2.3E-308 a 1.7E+308	Igual que Float pero con mayor precisión (hasta 15 digitos decimales) y mayor rango
long	4 bytes	-2,147,483,648 a 2,147,483,647	Guarda numeros enteros con mayor capacidad de almacenamiento que int
long double	10 bytes	3.4E-4932 to 1.1E+4932	Mucho mayor precisión en calculo decimal, incluso mayor que double (hasta 19 digitos decimales)

Facultad de Ingeniería de Sistemas (



Tipos de Datos



Las variables en C pueden ser declaradas en cuatro lugares del módulo del programa:

- 1. Fuera de todas las funciones del programa, son las llamadas variables globales, accesibles desde cualquier parte del programa.
- 2. Dentro de una función, son las llamadas **variables locales**, accesibles tan solo por la función en las que se declaran.
- 3. Como parámetros a la función, accesibles de igual forma que si se declararan dentro de la función.
- 4. Dentro de un bloque de código del programa, accesible tan solo dentro del bloque donde se declara. Esta forma de declaración puede interpretarse como una variable local del bloque donde se declara.

(Enrique Vicente Bonet Esteban, 2018)



Tipos de Datos



Ejemplo:

```
#include <stdio.h>
int suma; // Variable global, accesible desde cualquier parte del programa
void Sumar(int x) // Variable local declarada como parámetro, solo accesible en la función Sumar
     suma=suma+x;
     return;
int main() //Función principal del programa
     int valor=9; //Variables local, solo accesibles en el programa principal
     suma=10;
     Sumar(valor);
     printf("%d\n",suma);
return(0);
```







Operadores aritméticos en C:

	Operador
Suma	+
Resta	-
Multiplicación	*
División	/
Módulo	%
Incremento	++
Decremento	







Operadores lógicos en C:

	Operador
Not	!
Mayor	>
Mayor e igual	>=
Menor	<
Menor e igual	<=
Igual	==
Diferente	!=
Y (AND)	&&
O (OR)	П







```
// Comentarios
#include<stdio.h>
int main() {
    • • •
    • • •
       return 0;
```

```
stdio.h
Significa "standard input-output header"
main()
```

Es la función donde se escribe el programa principal. La ejecución de un programa siempre empieza con esta función.

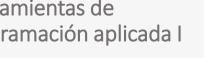




Estructura del lenguaje C

```
// Comentarios
#include<stdio.h>
#define PI 3.14
int main() {
   int x;
   float resultado;
   x=5;
   resultado = PI * x;
   printf("El resultado es: %.2f", resultado);
   return 0;
```







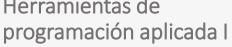
Entrada y salida de datos

La entrada y salida de datos se refiere a las operaciones que se producen en el teclado y la pantalla del ordenador. Dichos dispositivos son automáticamente abiertos y cerrados al comenzar y terminar el programa, por lo cual, no deben ser abiertos ni cerrados por el propio programa. Existen, básicamente, seis funciones de entrada y salida desde consola, tres de entrada y tres de salida.

La función scanf() se usa para leer cualquier tipo de dato predefinido desde el teclado, y convertirlo, de forma automática, al formato interno adecuado. La función se define como:







Entrada y salida de datos



La función scanf() se usa para leer cualquier tipo de dato predefinido desde el teclado, y convertirlo, de forma automática, al formato interno adecuado. La función se define como:

scanf ("%codigoFormato", &nombreVariableQueAlmacenaLaEntrada);

Código de formato	Significado
%d	Número positivo o negativo
%c	Un carácter (letra)
%s	Una cadena
%f	Un número con decimales
%e	Un número en notación científica



Entrada y salida de datos



Ejemplo con la función scanf()

```
#include <stdio.h>
// Ejemplo con la función scanf()

int main() {
        int valoNumerico1,valoNumerico2;
        printf("Introduzca el primer valor: \n");
        scanf("%d", &valoNumerico1);
        printf("Introduzca el segundo valor: \n");
        scanf("%d", &valoNumerico2);
        printf("Su valor 1 es %d y el valor 2 es %d",valoNumerico1,valoNumerico2);
    return 0;
}
```



programación aplic



Entrada y salida de datos

Práctica con la función scanf()

- 1. Escriba un programa que pida al docente 5 calificaciones para obtener el promedio final.
- 2. Escriba un programa que lea la temperatura en Fahrenheit y lo transforme a Celsius.
- 3. Escriba un programa que lea el precio de un producto y calcule el total a paga con el impuesto de 7%.
- 4. Escriba un programa que lea una cantidad en galones y lo transforme a litros.
- 5. Escriba un programa que lea el precio de un producto y le calcule el descuento del 25%. El programa debe imprimir el precio inicial, el descuento y el precio final.





Bibliografía

- Joyanes Aguilar, L. (2008). FUNDAMENTOS DE PROGRAMACIÓN Algoritmos, estructura de datos y objetos. Madrid: McGRAW-HILL/INTERAMERICANA DE ESPAÑA, S. A. U.
- Luis Joyanes, A., & Ignacio Zahonero, M. (2005). PROGRAMACIÓN EN C Metodología, algoritmos y estructura de datos. Madrid: Mc Graw Hill.
- Bonet Esteban, E. (2018). Lenguaje C.



@utpfisc







@utpfisc



fisc@utp.ac.pa





www.fisc.utp.ac.pa

