



TIPOS DE DATOS EN LENGUAJE C

Fundamentos de Programación

Peñaloza Castañeda Luna Quetzalli

Código de estudiante: 221618438

Maestra: Sánchez Rosario Patricia

Horario: martes y jueves 11:00 am – 12:55 pm

07/09/2024

NRC: 200274

Números enteros

Estos números en C se declaran con la palabra clave “int” y pueden tener opcionalmente dos tipos de prefijos modificadores:

- **“short” y “long”**: Estos modificadores ajustan el tamaño del entero en bits. Así, se tienen tres tipos de enteros: “int”, “short int” (abreviado como “short”) y “long int” (abreviado como “long”). El lenguaje C no establece tamaños específicos para sus tipos de datos básicos. Lo único que asegura es que un “short int” tiene un tamaño igual o menor que un “int”, y este último es igual o menor que un “long int”. Esta flexibilidad ha dificultado la creación de programas compatibles entre distintas plataformas.
- **“unsigned”**: Se utiliza para representar números naturales (mayores o iguales a cero)

Aquí un ejemplo de definición de un entero “int”:

```
int numero = 45;
```

Algo importante a mencionar es que los enteros, como funcionan en los conjuntos de números, pueden ser positivos, negativos o cero.

En el caso de la definición de un entero sin signo (números naturales):

```
unsigned int numero = 37;
```

La cantidad de bytes que ocupa un “int” puede variar según la plataforma y el compilador que utilices. Sin embargo, en la mayoría de los sistemas modernos, el tamaño típico de un int es de 4 bytes. Esto significa que puede almacenar valores en un rango bastante amplio, generalmente desde -2,147,483,648 hasta 2,147,483,647 para un “int” con signo.

Letras y cadenas

Las variables que almacenan caracteres individuales se declaran utilizando el tipo de dato “char” en lenguaje C. Para representar un carácter específica, este se debe escribir entre comillas simples.

Es importante recordar que internamente, los caracteres se codifican como números, siguiendo la tabla ASCII u otra codificación. Debido a esto es posible realizar operaciones aritméticas con los caracteres, como por ejemplo ‘M’ + 25, lo que sería equivalente a sumar el valor numérico correspondiente con la letra ‘M’ con 25.

Por otro lado, las cadenas de texto, también conocidas como “strings”, se tratan como arreglos o tablas de caracteres tipo “char”. En C, cada carácter en una cadena ocupa una posición en el arreglo, y las funciones estándar de la biblioteca de lenguaje para manipular asumen que la cadena termina cuando encuentran un byte

con valor cero (también conocido como carácter nulo o '\0'). Este carácter nulo actúa como un marcador que señala el final de la cadena.

En los programas escritos en C, las cadenas de texto se escriben rodeadas de comillas dobles, como, por ejemplo: "Hola, mundo". Además, *el lenguaje C añade automáticamente el valor cero al final de la cadena para garantizar que las funciones que las procesan sepan donde terminan.*

Ejemplos de declaración de un carácter individual y una cadena de texto:

Declaración de un carácter:

```
char letra = 'M';
```

Declaración de una cadena de texto:

```
char cadena[ ] = "Panchino";
```

El tamaño de un "char" es típicamente 1 byte. Esto se debe a que, en la mayoría de los sistemas, un byte está compuesto de 8 bits, y un "char" está diseñado para almacenar un solo carácter de texto, que, como mencionamos anteriormente, suele ser representado por un byte en el estándar ASCII.

Para las cadenas de texto, el tamaño depende de la longitud de la cadena más el carácter nulo que marca el final de cadena. Por ejemplo, la cadena "Hola" tiene un tamaño de 5 bytes: 4 bytes para los caracteres de la cadena y 1 byte adicional para el carácter nulo.

Números reales

Se declaran utilizando los tipos de datos "float" o "double". La diferencia principal entre ambos radica en la precisión que ofrecen debido a la forma en que se almacenan internamente. Dado que los números reales son infinitos, no pueden representarse en su totalidad con la limitada cantidad de bits de una computadora. Por tanto, cuanto mayor sea el número de bits que se utiliza para representar un número real, más números podrán ser representados y con mayor exactitud. En este sentido, los números reales declarados con "double" utilizan el doble de espacio en memoria que aquellos declarados con "float", lo que les permite ofrecer una mayor precisión.

Al igual que sucede con los enteros, el tamaño de las representaciones de los números reales no es uniforme entre diferentes plataformas. Esto significa que, dependiendo de la arquitectura o el sistema en el que se ejecute el programa, el tamaño en bits de un "float" o un "double" puede variar. Sin embargo, hay ciertos tamaños típicos que se suelen seguir: los números definidos como "float" suelen ocupar 4 bytes, mientras que los definidos como "double" ocupan generalmente 8 bytes.

Además, algunas plataformas proporcionan un tipo de número real con mayor precisión que “double”, conocido como “long double”. Estos números tienen un tamaño mayor que los “double” y son útiles en aplicaciones que requieren una exactitud extrema en los cálculos numéricos. Normalmente, los “long double” ocupan alrededor de 12 bytes, aunque esto también puede variar dependiendo de la plataforma.

Ejemplos de definiciones de estos números:

```
float n1 = 3.14f;
```

```
double n2 = 3.1415926535;
```

```
long double n3 = 3.141592653589793238L;
```

Booleanos

Es un tipo de dato muy utilizado en programación. Un valor booleano (bool) solo puede ser uno de dos: verdadero o falso, que internamente se representan como 1 (verdadero) y 0 (falso). En resumen, un booleano nos indica si algo es cierto o falso. En C, podemos usar las palabras “true” y “false” para representar estos valores.

Como el tipo bool fue añadido más tarde al lenguaje, es necesario incluir el archivo de cabecera “stdbool.h” para poder utilizarlo.

En muchas plataformas modernas y compiladores, el tamaño de un bool suele ser de 1 byte. Sin embargo, el estándar C no especifica un tamaño fijo para el tipo “bool”, solo que debe ser capaz de representar los valores “true” y “false”.

Bibliografía

1. *Booleanos en C - Parzibyte's blog*. (2018, noviembre 13). Parzibyte's blog; parzibyte. <https://parzibyte.me/blog/2018/11/12/booleanos-en-c/>
Recuperado el 7 de septiembre de 2024
2. Rancel, M. R. (s/f). *Tipos de datos en C. Declarar variables enteras int, long, o decimal float, double. char. Inicialización (CU00510F)*. Aprenderaprogramar.com; aprenderaprogramar.com, de https://www.aprenderaprogramar.com/index.php?option=com_content&view=article&id=899:tipos-de-datos-en-c-declarar-variables-enteras-int-long-o-decimal-float-double-char-inicializacion-cu00510f&catid=82&Itemid=210
Recuperado el 8 de septiembre de 2024
3. *Tipos de Datos en C*. (2013, marzo 28). La web de la programación. <https://duglasm.wordpress.com/tutoriales-de-progrmacion/tutorial-del->

[lenguaje-de-programacion-c/tipos-de-datos-en-c/](#)Recuperado el 7 de septiembre de 2024

4. Trini, S. (2021, mayo 5). Tipos de datos en C. *La35.net*. <https://la35.net/c/c-data-types.html> Recuperado el 8 de septiembre de 2024
5. Val, P. B., Ayres, I. E., & Pardo, A. (2018, julio 18). *Capítulo 2. Tipos de datos en C*. UC3M.
https://www.it.uc3m.es/pbasanta/asng/course_notes/data_types_es.html
Recuperado el 8 de septiembre de 2024