



Instituto Politécnico Nacional

Escuela Superior de Computo



Fundamentos De Programación

Pérez García Edgar Israel.

Arreglos y cadenas en C.

07 de octubre de 2025.

Introducción

En la programación, las variables simples como `int` o `char` son los ladrillos fundamentales que nos permiten almacenar y manipular datos individuales. Sin embargo, a medida que los problemas se vuelven más complejos, la necesidad de gestionar colecciones de datos relacionados se vuelve primordial. Un programa que solo puede manejar un número a la vez, o un único carácter, está severamente limitado. ¿Cómo podríamos, por ejemplo, almacenar las calificaciones de un grupo de estudiantes, los nombres en una agenda o simplemente un mensaje de texto para el usuario? La gestión de datos individuales para cada elemento sería impráctica y rápidamente inmanejable.

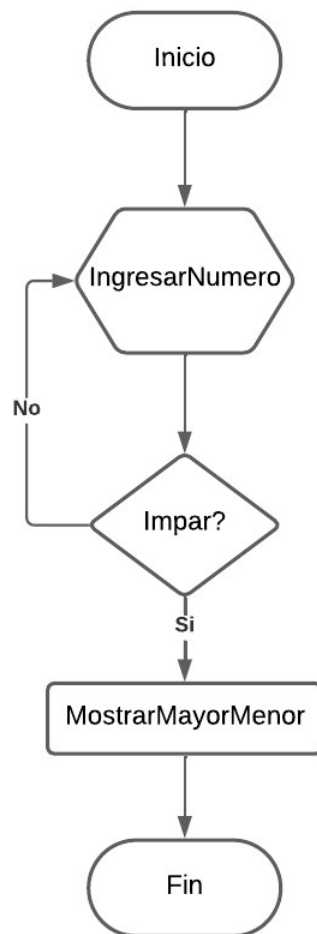
Para resolver esta necesidad fundamental, el lenguaje de programación C introduce el concepto de **arreglos** (arrays). Un arreglo es una estructura de datos que permite almacenar una colección de elementos del mismo tipo en una secuencia contigua de memoria, todos bajo un único nombre. Mediante el uso de un índice, podemos acceder, modificar y procesar cualquier elemento de la colección de manera eficiente, lo que nos permite pasar de manejar datos individuales a procesar conjuntos de datos de manera sistemática y ordenada. Los arreglos pueden ser unidimensionales, como una lista, o multidimensionales, como una tabla o una matriz, proporcionando una herramienta poderosa para organizar la información.

Un caso de uso particular y omnipresente de los arreglos es la manipulación de texto. En C, no existe un tipo de dato nativo para las cadenas de texto; en su lugar, se implementan como un **arreglo de caracteres** (`char`). Esta implementación tiene una convención fundamental: toda cadena debe terminar con un carácter especial nulo (`\0`), que actúa como un centinela para indicar el final del texto. Esta característica es crucial para que las funciones de la biblioteca estándar, como `printf()` o las contenidas en `string.h` (`strcpy`, `strlen`, etc.), puedan procesar las cadenas correctamente.

Dominar el uso de arreglos para datos numéricos y la implementación de cadenas como arreglos de caracteres es un paso esencial para cualquier programador. Permite pasar de la lógica simple que opera sobre valores únicos a la creación de programas robustos capaces de gestionar grandes volúmenes de información estructurada, sentando las bases para algoritmos de búsqueda, ordenamiento y manipulación de datos más avanzados.

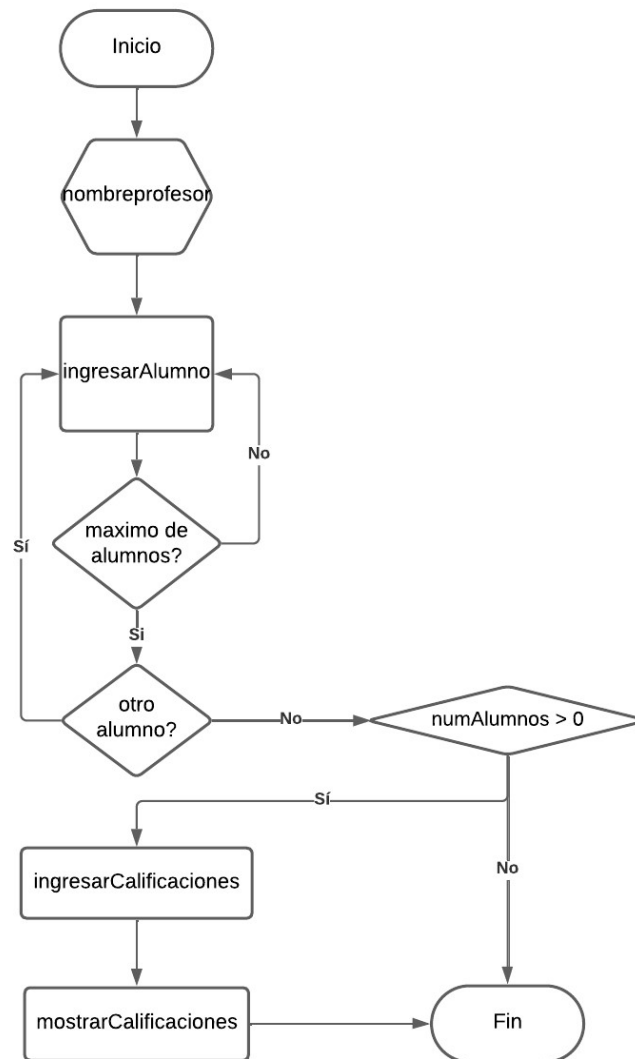
Desarrollo

1. Escriba un programa que lea valores de numeros enteros (tipo int) del usuario hasta que ingrese un número negativo, por ejemplo -21. Una vez que el usuario haya terminado de ingresar números, imprimir el valor más alto que haya ingresado, el valor más bajo que haya ingresado, y el número total de números que ingresó. El número negativo que se ingresa no debe tomarse como uno de los valores, solo es para detener la lectura de numeros del usuario. Utilice arreglos para resolver el problema, almacene como cadenas en un arreglo la solicitud de datos y la respuesta hacia el usuario cuando termine la ejecución.



El código realizado está en la misma carpeta que contiene este archivo con el nombre de PracticaCincoMenorMayorArreglos.c

2. Un profesor (almacenar su nombre en un arreglo) requiere de llevar las calificaciones de sus alumnos, necesita almacenar tareas, trabajos y exámenes, así como las calificaciones de cada rubro, sus correspondientes exámenes, su nombre y su boleta (almacenados en el mismo arreglo que el nombre del profesor). Emplee arreglos para hacer el trabajo. Al menos deberá almacenar hasta 10 trabajos por rubro calificado.



El código realizado está en la misma carpeta que contiene este archivo con el nombre de PracticaCincoAlumnos.c

3. En climas fríos, los meteorólogos reportan un índice llamado factor de sensación térmica, que tiene en cuenta la velocidad del viento y la temperatura. El índice proporciona una medida del efecto de enfriamiento del viento a una temperatura del aire dada. La sensación térmica se puede aproximar mediante la fórmula:

$$W = 13.12 + 0.6215 * t - 11.37 * v^{0.16} + 0.3965 * v^{0.16}$$

Donde:

v = velocidad del viento en km/h

t = temperatura en grados Celsius: $t \leq 10$

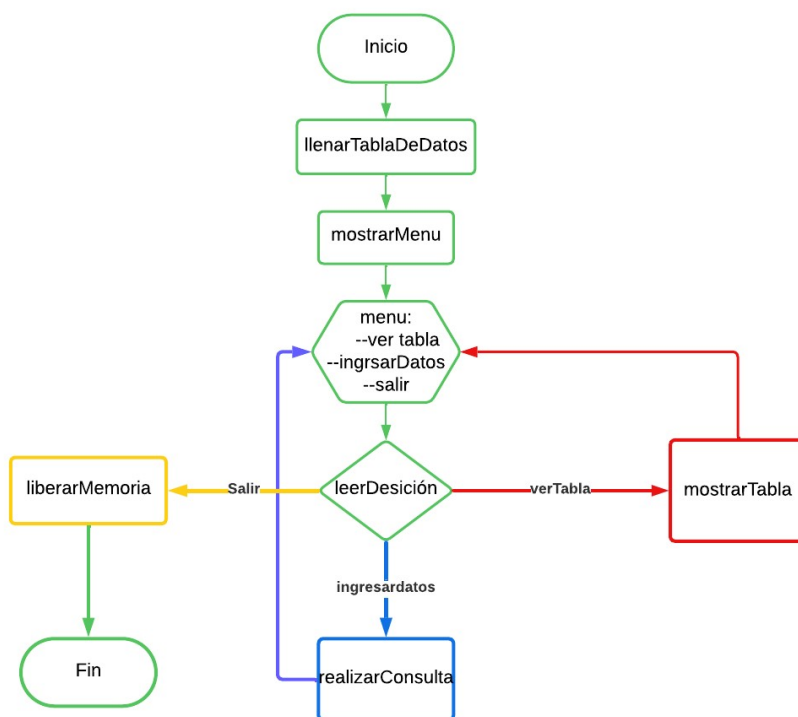
W = índice de sensación térmica (en grados Celsius)

- a) Cree en primer lugar la tabla.
- b) Cree un menú en donde se vean las siguientes opciones:
 1. Ver la tabla
 2. Realizar una consulta (deberá devolver también la respuesta del riesgo almacenados en un arreglo, de acuerdo a los límites mostrados en la tabla).
 - 2.1. EL programa muestra la tabla.
 - 2.2. El usuario introduce valores de temperatura y velocidad del viento, si estos valores están en la tabla consúltela y extraiga el valor de la celda correspondiente.
 - 2.3. En caso contrario calcule con la función del punto c.
- c) Escriba una función que devuelva el índice de sensación térmica.

La tabla deberá estar bien ajustada en cuanto a que el usuario la pueda ver sin problemas. Su código debe garantizar que no se viole la restricción de temperatura. Busque algunos informes meteorológicos en ediciones anteriores de un periódico en la biblioteca de su universidad y compare el índice de sensación térmica que calcule con el resultado informado en el periódico, emplee arreglos para almacenar la información calculada y cuando un usuario pregunte por el factor de congelación busque la información en la tabla (sin calcular) y devolver el mensaje del riesgo. Investigando un poco se encontró la siguiente tabla, genere la tabla empleando arreglos, verifique los valores que están en ella y comente al respecto. Si en otro lugar emplean otra ecuación para el cálculo de la sensación térmica y la cambia, justifique por qué.

Tabla Sensación térmica. (<https://concepto.de/sensacion-termica/>) 2-JUL-2023:

Viento (km/h)	10	7.5	5	2.5	0	-2.5	-5	-7.5	-10	-12.5	-15	-17.5	-20	-22.5	-25	-27.5	-30
8	7.5	5	2.5	0	-2.5	-5	-7.5	-10	-12.5	-15	-17.5	-20	-22.5	-25	-27.5	-30	-32.5
16	5	2.5	0	-2.5	-5	-7.5	-10	-12.5	-15	-17.5	-20	-25	-27.5	-32.5	-35	-37.5	-40
24	2.5	0	-5	-7.5	-10	-12.5	-17.5	-20	-25	-27.5	-32.5	-35	-37.5	-42.5	-45	-47.5	-52.5
32	0	-2.5	-7.5	-10	-12.5	-17.5	-22.5	-25	-30	-32.5	-35	-37.5	-42.5	-47.5	-50	-52.5	-57.5
40	0	-5	-7.5	-10	-15	-17.5	-22.5	-25	-30	-32.5	-37.5	-40	-45	-47.5	-52.5	-55	-60
48	-2.5	-5	-10	-12.5	-17.5	-20	-25	-27.5	-32.5	-35	-40	-42.5	-47.5	-50	-55	-57.5	-62.5
56	-2.5	-7.5	-10	-12.5	-17.5	-20	-25	-30	-30	-35	-42.5	-45	-50	-52.5	-55	-60	-67.5
64	-2.5	-7.5	-10	-15	-20	-22.5	-27.5	-30	-35	-37.5	-42.5	-45	-50	-55	-60	-62.5	-70



El código realizado está en la misma carpeta que contiene este archivo con el nombre de PracticaCincoTemperatura.c

Observaciones:

1. Se corroboró que la formula proporcionada era incorrecta. Se sustituyó por:

$$ST_{frio} = 13.12 + 0.6215 * T - 11.37 * V0.16 + 0.3965 * T * V0.16$$

Conclusiones.

Esta práctica ha sido fundamental para ilustrar la importancia de los arreglos y las cadenas como estructuras de datos esenciales en el lenguaje C. A través de los ejercicios, se ha demostrado que los arreglos son la herramienta principal para pasar del manejo de datos individuales a la gestión eficiente de colecciones de datos del mismo tipo, como listas de calificaciones o series numéricas.

Asimismo, se ha consolidado la comprensión de que las cadenas de texto en C no son un tipo de dato primitivo, sino una implementación específica de un arreglo de caracteres terminado con un carácter nulo (\0). Esta práctica ha permitido aplicar funciones de la biblioteca estándar para manipular tanto arreglos numéricos como cadenas, evidenciando que su correcto dominio es indispensable para desarrollar programas capaces de procesar conjuntos de información complejos y realizar tareas de entrada y salida de manera efectiva.

Referencias

Kernighan, B. W., & Ritchie, D. M. (2014). *The C Programming Language* (2nd ed.). Pearson Education.

Prata, S. (2013). *C Primer Plus* (6th ed.). Addison-Wesley Professional.

Deitel, P. J., & Deitel, H. M. (2016). *C How to Program* (8th ed.). Pearson.

King, K. N. (2008). *C Programming: A Modern Approach* (2nd ed.). W. W. Norton & Company.