

## RESUMEN cap2

- Los comentarios comienzan con `/*` y terminan con `*/`. Los programadores insertan comentarios para documentar sus programas y para mejorar su legibilidad. Los comentarios no ocasionan acción alguna cuando se ejecuta el programa.
- La directiva del preprocesador `#include<stdio.h>` le indica al compilador que incluya en el programa el encabezado estándar de entrada/salida. Este archivo contiene información que el compilador utiliza para verificar la precisión de las llamadas a funciones de entrada y salida, como `scanf` y `printf`.
- Los programas en C consisten en funciones, una de las cuales debe ser `main`. Todo programa en C comienza su ejecución en la función `main`.
- La función `printf` puede utilizarse para imprimir una cadena que se encuentra entre comillas, y para imprimir los valores de expresiones. Cuando se imprime un valor entero, el primer argumento de la función `printf` (la cadena de control de formato) contiene el especificador de conversión `%d` y cualquier otro carácter a imprimir; el segundo argumento es la expresión cuyo valor se imprimirá. Si se va a imprimir más de un entero, la cadena de control de formato contiene un `%d` para cada entero, y los argumentos separados por comas que siguen a la cadena de control de formato contienen las expresiones cuyos valores se imprimirán.
- La función `scanf` obtiene valores que el usuario normalmente introduce por medio del teclado. Su primer argumento es la cadena de control de formato que le indica a la computadora qué tipo de dato debe introducir el usuario. El especificador de conversión, `%d`, indica que el dato debe ser un entero. Cada uno de los argumentos restantes corresponden a uno de los especificadores de conversión de la cadena de control de formato. En general, todo nombre de variable es precedido por un ampersand (&), llamado operador de dirección. El ampersand, cuando se combina con el nombre de una variable, le indica a la computadora la posición de memoria en donde se almacenará el valor. Después la computadora almacena el valor en esa posición.
- Todas las variables deben declararse antes de que puedan utilizarse en un programa.
- Un nombre de variable es cualquier identificador válido. Un identificador es una serie de caracteres compuestos por letras, dígitos y guiones bajos (`_`). Los identificadores no deben comenzar con un dígito. Los identificadores pueden tener cualquier longitud, sin embargo, sólo los primeros 31 dígitos son importantes.
- C es sensible a mayúsculas y minúsculas.
- La mayoría de los cálculos se realizan en instrucciones de asignación.
- Toda variable almacenada en la memoria de la computadora tiene un nombre, un valor y un tipo.
- Siempre que un nuevo valor se coloque en una posición de memoria, éste reemplaza al valor anterior de esa posición. Debido a que la información anterior se destruye, al proceso de leer información en una posición de memoria se le conoce como lectura destructiva.
- Al proceso de lectura desde una posición de memoria se le conoce como lectura no destructiva.
- Las expresiones aritméticas deben escribirse en forma de línea recta, para facilitar la introducción de programas a la computadora.
- El compilador evalúa expresiones aritméticas en una secuencia precisa, determinada por las reglas de precedencia y de asociatividad de operadores.
- La instrucción `if` permite al programador tomar decisiones cuando se cumple cierta condición.
- Si la condición es verdadera, entonces se ejecuta la instrucción en el cuerpo de `if`. Si la condición es falsa, se salta la instrucción del cuerpo.
- Por lo general, las condiciones en instrucciones `if` se forman utilizando operadores de igualdad o de relación. El resultado de utilizar estos operadores siempre es la simple observación de “verdadero” o “falso”. Observe que las condiciones pueden ser expresiones que generen un valor cero (falso), o uno diferente de cero (verdadero).

### Complete los espacios en blanco:

- Al procedimiento para resolver un problema en términos de las acciones que se deben ejecutar y del orden en el que se deben ejecutar dichas órdenes se le llama .
- A la especificación del orden de ejecución de las instrucciones por medio de la computadora se le llama .
- Todos los programas pueden escribirse en términos de tres tipos de instrucciones de control: , y .
- La instrucción de selección se utiliza para ejecutar una acción cuando una condición es verdadera y otra acción cuando dicha condición es falsa.
- A muchas instrucciones agrupadas dentro de llaves (`{` y `}`), se les llama .
- La instrucción de repetición especifica que una instrucción o grupo de instrucciones se ejecutará de manera repetida mientras una condición sea verdadera.
- A la repetición de un conjunto de instrucciones, un número específico de veces se le llama repetición .
- Cuando no se sabe por adelantado el número de veces que se repetirá un conjunto de instrucciones, se puede utilizar un valor para terminar la repetición.

## RESUMEN cap3

- La solución de cualquier problema de computación involucra una serie de acciones en un orden específico. Al procedimiento para resolver un problema en términos de las acciones que se van a ejecutar y el orden en el que dichas acciones se deben ejecutar se le llama algoritmo.
- A la especificación del orden en el cual se van a ejecutar las instrucciones dentro de un programa se le llama control del programa.
- El pseudocódigo es un lenguaje artificial e informal que ayuda a los programadores a desarrollar algoritmos. Es similar al idioma inglés. En realidad, los programas en pseudocódigo no se ejecutan en las computadoras; solamente ayuda al programador a “plantear” un programa antes de intentar escribirlo en un lenguaje de programación tal como C.
- El pseudocódigo solamente consiste en caracteres, de manera que los programadores pueden teclear programas en pseudocódigo dentro de la computadora, editarlos, y guardarlos.
- El pseudocódigo consiste solamente en instrucciones ejecutables. Las declaraciones son mensajes para el compilador, para indicarle las características de las variables y reservar espacio para éstas.
- Una instrucción de selección se utiliza para elegir entre distintos cursos de acción.
- La instrucción de selección if ejecuta la acción indicada solamente si la condición es verdadera.
- La instrucción de selección if...else especifica la ejecución de acciones por separado: cuando la condición es verdadera y cuando la condición es falsa.
- Una instrucción de selección if...else anidada puede evaluar muchos casos diferentes. Si más de una condición es verdadera, solamente se ejecutarán las instrucciones que se encuentran después de la primera condición verdadera.
- Siempre que se vaya a ejecutar más de una instrucción, en donde por lo general se coloca sólo una, dichas instrucciones deben encerrarse entre llaves para formar una instrucción compuesta. Una instrucción compuesta puede colocarse en cualquier parte donde se pueda colocar una instrucción simple.
- Una instrucción vacía, que indica que no se realizará acción alguna, se establece mediante un punto y coma (;) en donde normalmente iría una instrucción.
- Una instrucción de repetición especifica que una acción se repetirá mientras cierta condición sea verdadera.
- La instrucción (o instrucción compuesta o bloque) contenida en la instrucción de repetición while constituye el cuerpo del ciclo.
- Por lo general, alguna instrucción especificada dentro del cuerpo de una instrucción while, en algún momento modificará la condición para que sea falsa. De lo contrario, el ciclo nunca terminará; un error conocido como ciclo infinito.
- El ciclo controlado por contador utiliza una variable como un contador para determinar cuándo debe terminar el ciclo.
- Un total es una variable que acumula la suma de una serie de números. Por lo general, los totales se inicializan en cero antes de la ejecución del programa.
- Un diagrama de flujo es una representación gráfica de un algoritmo. Los diagramas de flujo se dibujan utilizando ciertos símbolos especiales como óvalos, rectángulos, rombos, y pequeños círculos conectados mediante flechas llamadas líneas de flujo. Los símbolos indican las acciones a realizar. Las líneas de flujo indican el orden en el que se realizan las acciones.
- El símbolo óvalo, también llamado símbolo de terminación, indica el inicio y el final de cada algoritmo.
- El símbolo rectángulo, también llamado símbolo de acción, indica cualquier tipo de cálculo u operación de entrada/salida. Por lo general, los símbolos rectángulos corresponden a las acciones que realizan las instrucciones de asignación, o a las operaciones de entrada/salida que normalmente llevan a cabo funciones de la biblioteca estándar como printf y scanf.
- El símbolo rombo, también llamado símbolo de decisión, indica que se tomará una decisión. El símbolo de decisión contiene una expresión que puede ser falsa o verdadera. Dos líneas de flujo emergen de él. Una línea de flujo indica la dirección que se debe tomar cuando la condición es verdadera; la otra indica la dirección que se debe tomar cuando la condición es falsa.
- A un valor que contiene una parte fraccional se le conoce como número de punto flotante y se representa mediante el tipo de dato float.
- Cuando se dividen dos enteros, se pierde la parte fraccionaria del cálculo (es decir, se trunca).
- C proporciona el operador unario de conversión de tipo (float) para crear una copia de punto flotante de su operando. Al uso de un operador de conversión de tipo se le llama conversión explícita. Los operadores de conversión de flujo están disponibles para la mayoría de los tipos de datos.
- El compilador de C sabe cómo evaluar expresiones, sólo cuando los tipos de los operandos son idénticos. Para asegurarse de que los operandos sean del mismo tipo, el compilador realiza una operación llamada promoción (también conocida como conversión implícita) sobre los operandos seleccionados. En particular, los operandos

int se promueven a float. C proporciona un conjunto de reglas para la promoción de operandos de tipos diferentes.

- Los valores de punto flotante aparecen con un número específico de dígitos después del punto decimal cuando se especifica una precisión con el especificador de precisión %f dentro de una instrucción printf. El valor 3.456 aparece como 3.46 cuando se le aplica el especificador de conversión %.2f. Si utilizamos el especificador de conversión %f (sin especificar la precisión), se utiliza la precisión predeterminada 6.
- C proporciona varios operadores de asignación que ayudan a abreviar ciertos tipos comunes de expresiones de asignación. Estos operadores son: +=, -=, \*=, /=, y %=.

Variable = variable

operador expresión; donde operador es uno de los operadores +, -, \*, / o %, se puede escribir de la forma

Variable operador= expresión;

- C proporciona el operador de incremento, ++ y el operador de decremento --, para incrementar o decrementar una variable en 1. Estos operadores se pueden colocar como prefijo o sufijo de una variable. Si el operador se coloca como prefijo de la variable, ésta incrementa primero en 1, y luego se utiliza en su expresión. Si el operador

## RESUMEN cap4

- Un ciclo es un conjunto de instrucciones que la computadora ejecuta repetidamente, hasta que una condición de terminación se satisface. Dos formas de repetición son: la controlada por contador y la controlada por centinela.
  - Un contador de ciclo se utiliza para contar el número de veces que debe repetirse un grupo de instrucciones. Éste se incrementa (normalmente en 1) cada vez que el grupo de instrucciones se ejecuta.
  - Los valores centinela se utilizan generalmente para controlar una repetición en la que no se conoce por adelantado el número preciso de repeticiones, y el ciclo incluye instrucciones para obtener los datos cada vez que el ciclo se ejecuta.
  - Un valor centinela se introduce después de que todos los datos regulares se le han proporcionado al programa. Los centinelas deben elegirse cuidadosamente para que no exista posibilidad alguna de confundirlos con datos válidos.
  - La instrucción de repetición for maneja todos los detalles de la repetición controlada por contador. La forma general de la instrucción for es for ( expresión1; expresión2; expresión3 ) instrucción donde expresión1 inicializa la variable de control del ciclo, expresión2 es la condición de continuación del ciclo, y expresión3 incrementa (o decrementa) la variable de control.
  - La instrucción de repetición do...while es parecida a la instrucción de repetición while, pero la primera evalúa la condición de repetición de ciclo al final del ciclo, de tal forma que el ciclo se ejecutará al menos una vez. La forma de la instrucción do...while es do instrucción while ( condición );
  - La instrucción break, cuando se ejecuta en una de las instrucciones de repetición (for, while y do...while), ocasiona la salida inmediata de la instrucción. La ejecución continúa con la primera instrucción después del ciclo. La instrucción break también puede utilizarse para salir de una instrucción switch.
  - La instrucción continue, cuando se ejecuta en una de las instrucciones de repetición (for, while y do...while), salta cualquier instrucción restante del cuerpo de la instrucción de control, y continúa con la siguiente iteración del ciclo.
  - La instrucción switch maneja una serie de decisiones en las que una variable o expresión en particular se evalúa con cada uno de los valores que puede asumir, y se toman diferentes acciones. Cada case de una instrucción switch puede ocasionar que se ejecuten muchas instrucciones. En la mayoría de los programas es necesario incluir un break después de las instrucciones de cada case, de otro modo, el programa ejecutará las instrucciones de cada case hasta que encuentre un break, o hasta que alcance el final de la instrucción switch. Muchos cases pueden ejecutar las mismas instrucciones, listando las etiquetas case antes de las instrucciones. La instrucción switch sólo puede evaluar expresiones integrales constantes.
  - La función getch devuelve un carácter proveniente del teclado (la entrada estándar) como un entero.
  - En sistemas UNIX y en muchos otros, el carácter EOF se introduce escribiendo <Entrar> <Control+d>
- En sistemas Windows de Microsoft, el carácter EOF se introduce escribiendo <Control+z>
- Los operadores lógicos pueden utilizarse para formar condiciones complejas, mediante la combinación de condiciones. Los operadores lógicos son &&, || y !, que significan AND lógico, OR lógico y NOT lógico (negación), respectivamente.
  - Un valor verdadero es cualquier valor diferente de cero.
  - Un valor falso es 0 (cero).

### Complete los espacios en blanco.

- a) A la repetición controlada por contador también se le conoce como repetición **definida**, ya que se sabe por adelantado el número de veces que se ejecutará el ciclo.
- b) A la repetición controlada por centinela también se le conoce como repetición **Indefinida**, ya que no se sabe por adelantado el número de veces que se ejecutará el ciclo.
- c) En la repetición controlada por contador se utiliza un **variable de control o contador** para contar el número de veces que un grupo de instrucciones debe repetirse.
- d) La instrucción **continue**, cuando se ejecuta en una instrucción de repetición, ocasiona que se ejecute inmediatamente la siguiente iteración del ciclo.
- e) La instrucción **break**, cuando se ejecuta en una instrucción de repetición o en un switch, ocasiona la salida inmediata de la instrucción.
- f) La **instrucción de selección** se utiliza para evaluar una variable o expresión en particular para cada uno de los valores integrales constantes que puede asumir.

**Diga si los siguientes enunciados son verdaderos o falsos. Si la respuesta es falso, explique por qué.**

- a) En la instrucción de selección switch, es necesario un caso default.

**Falso.** El caso default es opcional. Si no es necesaria una acción predeterminada, entonces no se necesita un caso default.

- b) La instrucción break es necesaria en el caso default de una instrucción de selección switch.

**Falso.** La instrucción break se utiliza para salir de la instrucción switch. La instrucción break no es necesaria cuando el caso default es el último caso.

- c) La expresión  $(x > y \ \&\& \ a < b)$  es verdadera si  $x > y$  o si  $a < b$ .

**Falso.** Cuando se utiliza el operador  $\&\&$ , ambas expresiones de relación deben ser verdaderas para que toda la expresión sea verdadera.

- d) Una expresión que contiene el operador  $\|\$  es verdadera si uno o ambos de sus operandos son verdaderos.

**Verdadero**

### PALABRAS RESERVADA: cap2

**auto:** Una palabra clave que se utiliza para declarar variables locales que son automáticas, es decir, se crean y se destruyen automáticamente en función del ámbito en el que se encuentren.

**break:** Se utiliza dentro de bucles y estructuras de control switch para salir de inmediato de la estructura en la que se encuentra.

**case:** Utilizado dentro de una estructura de control switch para definir posibles valores coincidentes para la evaluación de una expresión.

**char:** Un tipo de dato que representa un único carácter ASCII o un número entero pequeño.

**const:** Una palabra clave que se utiliza para declarar variables cuyos valores no pueden ser modificados después de la inicialización.

**continue:** Utilizado dentro de bucles para interrumpir la iteración actual y pasar a la siguiente iteración.

**default:** Utilizado dentro de una estructura de control switch para proporcionar un caso por defecto que se ejecutará si ninguno de los casos coincidentes es válido.

**do:** Utilizado para iniciar un bucle do...while, que ejecutará un bloque de código al menos una vez y luego verificará una condición.

**double:** Un tipo de dato que representa números de punto flotante con mayor precisión que el tipo float.

**else:** Utilizado en estructuras de control if o if...else para especificar un bloque de código que se ejecutará si la condición no es verdadera.

**enum:** Utilizado para declarar un tipo de dato enumerado, que representa un conjunto de constantes enteras relacionadas.

**extern:** Utilizado para declarar una variable o función que está definida en otro lugar, fuera del archivo actual.

**float:** Un tipo de dato que representa números de punto flotante de precisión simple.

**for:** Utilizado para iniciar un bucle for, que se repite mientras una condición es verdadera y se controla a través de una variable de control.

**goto:** Una palabra clave que se utiliza para transferir el control a una etiqueta específica en el código.

**if:** Utilizado para iniciar una estructura de control if, que ejecuta un bloque de código si la condición especificada es verdadera.

**int:** Un tipo de dato que representa números enteros.

**long:** Un modificador que se utiliza con tipos de datos enteros para indicar que son "largos".

**register:** Una palabra clave que se utiliza para sugerir al compilador que almacene la variable en un registro del procesador para un acceso más rápido.

**return:** Utilizado dentro de una función para devolver un valor y salir de la función.

**TERMINOLOGIA cap2**

**Acción:** Un conjunto de instrucciones ejecutadas como una unidad en un programa.

**Ampersand (&):** Un operador utilizado para obtener la dirección de una variable en memoria.

**Argumento:** Un valor proporcionado a una función cuando se llama, que puede ser utilizado por la función para realizar una tarea.

**Asociatividad de Derecha a Izquierda:** La forma en que los operadores con la misma precedencia se evalúan de derecha a izquierda.

**Asociatividad de Izquierda a Derecha:** La forma en que los operadores con la misma precedencia se evalúan de izquierda a derecha.

**Asociatividad de Operadores:** El orden en que se evalúan los operadores cuando hay varios operadores en una expresión.

**Asterisco (\*):** Un operador utilizado para declarar un puntero o para desreferenciar un puntero.

**Biblioteca Estándar de C:** Una colección de funciones y macros que proporciona funcionalidades básicas en C, como entrada/salida, manejo de cadenas, matemáticas, entre otros.

**Bloque:** Un conjunto de instrucciones agrupadas entre llaves {}. Puede ser una función, un bucle, una estructura de control, etc.

**Cadena:** Una secuencia de caracteres en C, generalmente representada como un arreglo de caracteres terminado por un carácter nulo \0.

**Cadena de Caracteres:** Un conjunto de caracteres alfanuméricos que forman una secuencia en un programa.

**Cadena de Control:** Una cadena de caracteres utilizada como argumento en funciones como printf para formatear la salida.

**Cadena de Control de Formato:** Una cadena de control utilizada en funciones como printf para especificar cómo se debe formatear la salida.

**Carácter de Escape:** Un carácter especial precedido por la barra invertida \, que tiene un significado especial en cadenas y literales.

**Carácter de Escape Diagonal Invertida ():** El carácter que se utiliza para indicar el inicio de un carácter de escape.

**Carácter de Nueva Línea (\n):** Un carácter de escape que representa una nueva línea en una cadena o en un flujo de salida.

**Carácter Espacio en Blanco:** Cualquier carácter que representa un espacio vacío en una cadena o en un flujo de entrada.

**Comentario:** Texto en el código fuente que no se compila ni se ejecuta y se utiliza para hacer notas o explicaciones para los programadores.

**Computación Conversacional:** Un tipo de interacción entre un usuario y una computadora en el que el usuario ingresa comandos y la computadora responde en tiempo real.

**Computación Interactiva:** Un proceso en el que los usuarios interactúan con una computadora para proporcionar entrada y recibir resultados en tiempo real.

**Condición:** Una expresión booleana que determina si una instrucción o conjunto de instrucciones se ejecutará.

**Cuerpo de una Función:** El conjunto de instrucciones que se ejecutan cuando se llama a una función.

**Decisión:** El proceso de elegir entre varias alternativas en un programa en función de una condición.

**Declaración:** Una instrucción que le dice al compilador el tipo y nombre de una variable o función antes de usarla.

**División Entera:** Una operación aritmética que produce solo la parte entera del resultado de la división.

**División entre Cero:** Una operación que resulta en un error cuando intentas dividir por cero.

**Encabezado Estándar de Entrada/Salida:** Un archivo de encabezado (<stdio.h>) que contiene declaraciones para funciones de entrada/salida estándar.

**Entero:** Un tipo de dato que representa números enteros positivos o negativos.

**Error de Compilación:** Un error que ocurre durante la etapa de compilación debido a una construcción incorrecta en el código fuente.

**Error de Sintaxis:** Un error que ocurre cuando el código fuente no sigue las reglas gramaticales del lenguaje.

**Error en Tiempo de Compilación:** Un error que ocurre durante la etapa de compilación debido a problemas en la estructura del programa.

**Error Fatal:** Un error que causa la terminación inmediata y anormal de un programa.

**Error No Fatal:** Un error que no interrumpe la ejecución del programa pero puede afectar su comportamiento.

**Especificador de Conversión:** Un carácter utilizado en una cadena de control para indicar el tipo de dato que se debe formatear.

**Especificador de Conversión %d:** Utilizado en una cadena de control para formatear un valor entero.

**Falso:** El valor booleano que indica que una condición no se cumple o que una expresión es falsa.

**Flujo de Control:** La secuencia en la que las instrucciones se ejecutan en un programa.

**Forma de Línea Recta:** Un flujo de control directo y secuencial sin bifurcaciones ni repeticiones.

**Función:** Un bloque de código que realiza una tarea específica. Puede aceptar argumentos y devolver un valor.

**Función printf:** Una función estándar en C utilizada para imprimir datos formateados en la consola.

**Función scanf:** Una función estándar en C utilizada para leer datos desde la consola.

**Guión Bajo (\_):** Un carácter que se puede usar en nombres de variables y funciones, generalmente para separar palabras.

**Identificador:** Un nombre que se utiliza para identificar una variable, función o cualquier otro elemento en un programa.

**Indicador:** Un carácter especial utilizado en cadenas de control de formato para especificar el tipo de dato que se debe formatear.

**Instrucción:** Una operación básica en un programa que se realiza secuencialmente.

**Instrucción de Asignación:** Una instrucción que asigna un valor a una variable utilizando el operador de asignación =.

**Instrucción de Control if:** Una instrucción que permite que se ejecute un bloque de código si una condición es verdadera.

**Int:** Un tipo de dato que representa números enteros.

**Lectura No Destructiva:** La acción de leer datos sin alterarlos en la memoria.

**Literal:** Un valor constante que aparece directamente en el código fuente.

**Llaves {}:** Símbolos utilizados para delimitar el alcance de un bloque de código.

**Main:** La función principal en un programa C que se ejecuta primero y desde donde comienza la ejecución.

**Memoria:** El espacio donde se almacenan los datos y las instrucciones de un programa.

**Mensaje:** Un texto que se muestra en la consola o en la interfaz de usuario para proporcionar información al usuario.

**Modelo de Acción/Decisión:** Una forma de diseñar un programa que involucra tomar decisiones basadas en condiciones.

**Nombre:** Un identificador utilizado para referirse a una variable, función u otro elemento en el programa.

**Nombre de Variable:** Un identificador utilizado para referirse a una ubicación en memoria que almacena un valor.

**Operador:** Un símbolo que realiza una operación en uno o más operandos para producir un resultado.

**Operador de Asignación (=):** Un operador que asigna un valor a una variable.

**Operador de Asignación Signo de Igual (=):** Un operador utilizado para asignar un valor a una variable.

**Operador de Dirección:** Un operador que devuelve la dirección en memoria de una variable.

**Operador de Multiplicación (\*):** Un operador utilizado para realizar la multiplicación entre dos valores.

**Operador Módulo (%):** Un operador utilizado para calcular el residuo de la división entre dos valores.

**Operadores Aritméticos:** Operadores utilizados para realizar operaciones matemáticas básicas, como suma, resta, multiplicación y división.

**Operadores Binarios:** Operadores que trabajan con dos operandos, como los operadores aritméticos y de relación.

**Operadores de Igualdad:** Operadores utilizados para comparar si dos valores son iguales o diferentes.

**== “Es Igual Que”:** Un operador de igualdad que compara si dos valores son iguales.

**!= “No Es Igual Que”:** Un operador de igualdad que compara si dos valores no son iguales.

**Operadores de Relación:** Operadores utilizados para comparar si un valor es mayor, menor, mayor o igual, o menor o igual que otro valor.

**> “Es Mayor Que”:** Un operador de relación que compara si un valor es mayor que otro.

**< “Es Menor Que”:** Un operador de relación que compara si un valor es menor que otro.

**>= “Es Mayor o Igual Que”:** Un operador de relación que compara si un valor es mayor o igual que otro.

**<= “Es Menor o Igual Que”:** Un operador de relación que compara si un valor es menor o igual que otro.

**Operando:** Un valor en una expresión que es utilizado por un operador para realizar una operación.

**Palabras Clave:** Palabras reservadas que tienen un significado especial en el lenguaje de programación y no pueden ser utilizadas como nombres de variables u otros identificadores.

**Palabras Reservadas:** Palabras que tienen un significado predefinido en el lenguaje de programación y se utilizan para realizar tareas específicas.

**Palabras Reservadas de C:** Palabras clave específicas que tienen significado en el lenguaje de programación C.

**Paréntesis ( ):** Símbolos utilizados para agrupar expresiones y controlar el orden de evaluación.

**Paréntesis Anidados:** Paréntesis que se encuentran dentro de otros paréntesis, utilizados para agrupar expresiones más complejas.

**Posición, Ubicación:** La dirección en memoria donde se almacenan datos o instrucciones.

**Precedencia:** La prioridad relativa entre diferentes operadores en una expresión.

- Preprocesador de C:** Una fase del proceso de compilación que realiza transformaciones en el código fuente antes de la compilación real.
- Programación Estructurada:** Un enfoque de programación que se basa en el uso de estructuras de control para organizar el flujo de ejecución.
- Reglas de Precedencia de Operadores:** Las reglas que determinan el orden en que se evalúan los operadores en una expresión.
- Sangría:** Espacios o tabulaciones al comienzo de una línea de código para mejorar la legibilidad y la estructura.
- Secuencia de Escape:** Una combinación de caracteres que representa un carácter especial en una cadena.
- Sensible a Mayúsculas y Minúsculas:** Una característica en la que el lenguaje distingue entre letras mayúsculas y minúsculas en los nombres de identificadores.
- Signo de Porcentaje (%) para Iniciar un Especificador de Conversión:** Un símbolo utilizado en cadenas de control de formato para indicar que se inicia un especificador de conversión.
- Stdio.h:** El archivo de encabezado estándar que proporciona funciones de entrada/salida en C.
- Tecla de Retorno:** La tecla utilizada para mover el cursor al principio de una línea de texto o para confirmar una entrada.
- Tecla Entrar:** La tecla utilizada para enviar una entrada y avanzar a la siguiente línea.
- Terminador de Instrucción (;):** Un carácter utilizado al final de una instrucción en C para indicar su final.
- Terminador de Instrucción Punto y Coma (;):** Un carácter utilizado para separar y delimitar instrucciones en C.
- Tipo de Variable:** El tipo de dato que una variable puede contener, como int, float, char, etc.
- Toma de Decisiones:** La capacidad de un programa para seleccionar un camino de ejecución en función de una condición.
- Ubicación (o Posición) de Memoria:** La dirección única en la memoria donde se almacena un valor o una instrucción.
- Valor:** Un dato que se almacena en una variable o se utiliza en una expresión.
- Valor Cero (Falso):** El valor numérico que representa falso o no verdadero en C.
- Valor de Variable:** El contenido almacenado en una variable en un momento dado.
- Valor Diferente de Cero (Verdadero):** Cualquier valor numérico distinto de cero se considera verdadero en C.
- Variable:** Un espacio en memoria que almacena

Complete los espacios en blanco: CP3

- a) Al procedimiento para resolver un problema en términos de las acciones que se deben ejecutar y del orden en el que se deben ejecutar dichas órdenes se le llama **algoritmo** .
- b) A la especificación del orden de ejecución de las instrucciones por medio de la computadora se le llama **Control de programa** .
- c) Todos los programas pueden escribirse en términos de tres tipos de instrucciones de control: **Secuencia, selección y repetición.**
- d) La instrucción de selección **if...else** se utiliza para ejecutar una acción cuando una condición es verdadera y otra acción cuando dicha condición es falsa.
- e) A muchas instrucciones agrupadas dentro de llaves ({ y }), se les llama **Instrucción compuesta.**
- f) La instrucción de repetición **while** especifica que una instrucción o grupo de instrucciones se ejecutará de manera repetida mientras una condición sea verdadera.
- g) A la repetición de un conjunto de instrucciones, un número específico de veces se le llama repetición **Controlada por contador.**
- h) Cuando no se sabe por adelantado el número de veces que se repetirá un conjunto de instrucciones, se puede utilizar un valor **centinela** para terminar la repetición.

### Terminologia cap3

- Acción:** Una instrucción individual o una operación realizada en un programa.
- Algoritmo:** Una serie ordenada de pasos o instrucciones diseñados para resolver un problema o realizar una tarea específica.
- Bloque:** Un grupo de declaraciones o instrucciones agrupadas entre llaves {} en C.

**Caracteres Blancos:** Espacios, tabulaciones y caracteres de nueva línea que se utilizan para dar formato y mejorar la legibilidad del código.

**Ciclo Infinito:** Un ciclo que se ejecuta repetidamente sin una condición de finalización, lo que puede llevar a que el programa se quede atrapado.

**Cima:** Un punto de control dentro de un programa, generalmente utilizado para dirigir la ejecución hacia una sección específica.

**Condición de Terminación:** Una expresión booleana utilizada para determinar cuándo debe finalizar un ciclo.

**Contador:** Una variable utilizada para realizar un seguimiento del número de iteraciones en un ciclo.

**Control de Programa:** La capacidad de guiar la ejecución del programa a través de estructuras de control como ciclos y decisiones.

**Conversión Explícita:** Cambiar el tipo de una variable de manera deliberada utilizando operadores de conversión, como (int).

**Conversión Implícita:** La conversión automática de un tipo de dato a otro sin la necesidad de operadores de conversión.

**Cuerpo de un Ciclo:** El conjunto de instrucciones que se ejecutan repetidamente mientras se cumple la condición del ciclo.

**Decisión:** Una estructura de control que permite que un programa tome diferentes caminos basados en condiciones.

**Diagrama de Flujo:** Una representación gráfica de un algoritmo o proceso, con símbolos que representan acciones, decisiones y flujo de control.

**División Entera:** Una operación que devuelve solo la parte entera del resultado de la división entre dos números enteros.

**División entre Cero:** Un error que ocurre cuando intentas dividir un número por cero, lo cual no es matemáticamente válido.

**Ejecución Secuencial:** La ejecución de instrucciones en el orden en que aparecen en el programa, una tras otra.

**Eliminación de Goto:** La práctica de evitar el uso de la instrucción goto en favor de estructuras de control más estructuradas.

**Error de Sintaxis:** Un error que ocurre cuando el código no sigue las reglas gramaticales del lenguaje y no se puede compilar.

**Error Fatal:** Un error que impide que el programa continúe ejecutándose y generalmente resulta en la terminación del programa.

**Error Lógico:** Un error en el cual el programa produce resultados incorrectos debido a una lógica incorrecta en el código.

**Errores no Fatales:** Errores que no detienen la ejecución del programa, pero pueden afectar el comportamiento o resultados.

**"Estallamiento":** Un error que ocurre cuando el resultado de una operación excede el rango permitido para el tipo de dato.

**"Estrellamiento":** Un error que ocurre cuando una operación se realiza con operandos incorrectos o incompatibles.

**Estructura de Control:** Un elemento de programación que controla el flujo de ejecución en un programa, como ciclos y decisiones.

**Estructura de Control de Entrada/Salida Simple:** Estructuras utilizadas para realizar operaciones básicas de entrada/salida, como printf y scanf.

**Estructura de Secuencia:** Un diseño de programación en el que las instrucciones se ejecutan en orden secuencial.

**Estructuras de Control Anidadas:** Estructuras de control dentro de otras estructuras de control.

**Estructuras de Control Apiladas:** Estructuras de control anidadas una dentro de otra en diferentes niveles de indentación.

**Estructuras de Repetición:** Estructuras que permiten la ejecución repetida de un bloque de código mientras se cumple una condición.

**Estructuras de Selección:** Estructuras que permiten tomar decisiones basadas en condiciones booleanas.

**Expresión Condicional:** Una expresión que se evalúa como verdadera o falsa, utilizada en decisiones y estructuras de control.

**Fase de Inicialización:** La etapa en la que las variables se definen y se les asignan valores iniciales.

**Fase de Procesamiento:** La etapa en la que se realizan cálculos y se ejecutan las instrucciones del programa.

**Fase de Terminación:** La etapa en la que se realizan operaciones de limpieza y finalización antes de que el programa termine.

**"Fin de la Entrada de Datos":** Una marca especial utilizada para indicar el final de la entrada de datos, a menudo en flujos de entrada.



**Float:** Un tipo de dato en C que representa números de punto flotante con precisión limitada.

**Inicialización:** Asignar un valor inicial a una variable cuando se declara.

**Instrucción Compuesta:** Una serie de instrucciones agrupadas entre llaves {} que se consideran una sola unidad de ejecución.

**Instrucción de Repetición While:** Una estructura de control que repite un bloque de código mientras se cumple una condición.

**Instrucción de Selección Doble:** Una estructura de control que permite tomar una de dos decisiones basadas en una condición.

**Instrucción de Selección If:** Una estructura de control que permite tomar decisiones basadas en una condición.

**Instrucción de Selección If...Else:** Una estructura de control que permite tomar decisiones alternativas basadas en una condición.

**Instrucción de Selección Múltiple:** Una estructura de control que permite tomar decisiones entre múltiples opciones basadas en una expresión.

**Instrucción de Selección Simple:** Una estructura de control que permite tomar una decisión basada en una condición.

**Instrucción Goto:** Una instrucción que permite saltar a una etiqueta específica en el programa.

**Instrucción Vacía (;):** Una instrucción que no realiza ninguna acción y se utiliza para separar declaraciones en C.

**Instrucciones If...Else Anidadas:** Instrucciones if...else colocadas dentro de otras instrucciones if...else.

**Línea de Flujo:** La secuencia ordenada de instrucciones que se ejecutan en un programa.

**Mejoramiento Arriba Abajo, Paso a Paso:** Un enfoque en la resolución de problemas que comienza con una descripción general y se desglosa en pasos detallados.

**Mejoramiento Paso a Paso:** Un enfoque en la resolución de problemas que implica dividir un problema en pasos más pequeños y manejables.

**Número de Punto Flotante:** Un tipo de dato que representa números con decimales con precisión limitada.

**Operador Condicional (?):** Un operador ternario que permite tomar decisiones en una sola línea de código.

**Operador de Conversión de Tipo:** Un operador que convierte un tipo de dato en otro, como (int) o (float).

**Operador de Decremento (--):** Un operador que disminuye el valor de una variable en 1.

**Operador de Incremento (++):** Un operador que aumenta el valor de una variable en 1.

**Operador de Posdecremento:** Un operador que primero utiliza el valor actual de una variable y luego disminuye su valor en 1.

**Operador de Posincremento:** Un operador que primero utiliza el valor actual de una variable y luego aumenta su valor en 1.

**Operador de Predecremento:** Un operador que disminuye el valor de una variable en 1 y luego utiliza su valor.

**Operador de Preincremento:** Un operador que aumenta el valor de una variable en 1 y luego utiliza su valor.

**Operador Ternario:** Un operador condicional que permite tomar decisiones en una sola línea de código.

**Operadores Aritméticos de Asignación:** Operadores que realizan una operación aritmética y asignan el resultado a una variable, como +=, -=, \*=, /=, y %=.

**Operadores de Multiplicación:** Operadores que realizan una multiplicación entre dos valores, como \*.

**Orden de las Acciones:** La secuencia en la que se ejecutan las instrucciones en un programa.

**Pasos:** Acciones individuales en un algoritmo o programa.

**Precisión:** La cantidad de dígitos significativos en un número de punto flotante.

**Precisión Predeterminada:** El número de dígitos significativos en un número de punto flotante que se utiliza por defecto.

**Primer Mejoramiento:** Un enfoque en la resolución de problemas que implica implementar la primera solución que parece funcionar.

**Programación Estructurada:** Un estilo de programación que utiliza estructuras de control para organizar y controlar la ejecución del programa.

**Promoción:** La conversión automática de un tipo de dato a un tipo más grande cuando se realizan operaciones.

**Pseudocódigo:** Una descripción en lenguaje humano de un algoritmo o programa que es independiente de un lenguaje de programación específico.

**Redondeo:** Ajustar un número a un valor cercano que tenga menos dígitos decimales.

**Repetición:** Ejecución repetida de un bloque de código.

**Repetición Controlada por Contador:** Una estructura de repetición que repite un bloque de código un número específico de veces.

**Repetición Definida:** Una estructura de repetición que repite un bloque de código mientras se cumple una condición.

**Repetición Indefinida:** Una estructura de repetición que repite un bloque de código hasta que se cumpla una condición de finalización.

**Segundo Mejoramiento:** Un enfoque en la resolución de problemas que implica revisar y mejorar una solución inicial.

**Selección:** Tomar decisiones en el flujo de ejecución del programa.

**Símbolo de Acción:** Un símbolo en un diagrama de flujo que representa una acción o instrucción.

**Símbolo de Decisión:** Un símbolo en un diagrama de flujo que representa una decisión.

**Símbolo de Diagrama de Flujo:** Un símbolo gráfico utilizado en un diagrama de flujo para representar diferentes elementos y acciones.

**Símbolo de Fin:** Un símbolo en un diagrama de flujo que indica el final del proceso.

**Símbolo de Óvalo:** Un símbolo en un diagrama de flujo que representa el inicio o fin de un proceso.

**Símbolo de Terminación:** Un símbolo en un diagrama de flujo que indica el final del proceso.

**Símbolo Rectángulo:** Un símbolo en un diagrama de flujo que representa una instrucción o acción.

**Símbolo Rombo:** Un símbolo en un diagrama de flujo que representa una decisión.

**Símbolos Conectores:** Símbolos en un diagrama de flujo que conectan diferentes partes del flujo de control.

**Símbolos de Flecha:** Símbolos utilizados en un diagrama de flujo para indicar la dirección del flujo de control.

**Total:** El resultado de una operación aritmética o una suma.

**Transferencia de Control:** Cambiar el flujo de ejecución del programa de un lugar a otro, como con la instrucción goto.

**Truncar:** Cortar la parte decimal de un número de punto flotante para convertirlo en un número entero.

**Valor "Basura":** Un valor no deseado o impredecible almacenado en una variable no inicializada.

**Valor Centinela:** Un valor especial utilizado para indicar el final de una secuencia de datos, por ejemplo, en la entrada de datos.

**Valor de Bandera:** Un valor que se utiliza para indicar una condición específica en el programa.

**Valor de Señal:** Un valor especial que indica una situación excepcional o error en el programa.

**Valor Falso:** Un valor booleano que representa una afirmación falsa.