


Programación en

PROGRAMACIONATS

LIBRO PRÁCTICO DE PROGRAMACIÓN EN C



Ejercicios Propuestos y Resueltos | Alejandro M. Taboada

Nivel 1: Operadores y Expresiones:

1. Pedir 2 números al usuario y sumarlos, restarlos, multiplicarlos y dividirlos.
2. Convertir Grados Celsius a Grados Fahrenheit.
3. Sacar la hipotenusa de un triángulo rectángulo, pidiendo al usuario el valor de los 2 catetos.
4. Hacer un Programa que calcule longitudes de Circunferencia.
5. Hacer un Programa que calcule áreas de trapecios.
6. Calcule la media aritmética de 3 números cualesquiera.
7. Una tienda ofrece un descuento del 15% sobre el total de la compra y un cliente desea saber cuánto deberá pagar finalmente por su compra.
8. Dadas las horas trabajadas de una persona y el valor por hora. Calcular su salario e imprimirlo.
9. Calcular el nuevo salario de un obrero, si obtuvo un incremento del 25% sobre su salario anterior.
10. Un alumno desea saber cuál será su calificación final en la materia de Algoritmos. Dicha calificación se compone de los siguientes porcentajes:
 - 55% del promedio de sus tres calificaciones parciales.
 - 30% de la calificación del examen final.
 - 15% de la calificación de un trabajo final.
11. Calcular la cantidad de segundos que están incluidos en el número de horas, minutos y segundos ingresados por el usuario.

Nivel 2: Estructuras de Selección – Condicionales:

1. Comprobar a través de un programa si un alumno aprobó o no un examen (Aprueba si su nota es mayor a 10.5).
2. Comprobar si un número digitado por el usuario es positivo o negativo.
3. Visualizar la tarifa de la luz según el gasto de corriente eléctrica. Para un gasto menor de 1.000Kwxh la tarifa es 1.2, entre 1.000y 1.850Kwxh es 1.0 y mayor de 1.850Kwxh 0.9.
4. Determinar si un número es par, impar.
5. Calcular el mayor de dos números leídos del teclado y visualizarlo en pantalla.
6. Ingrese un número y calcule e imprima su raíz cuadrada. Si el número es negativo imprima el número y un mensaje que diga "tiene raíz imaginaria".
7. Ingresar por teclado el nombre y el signo de cualquier persona e imprima, el nombre solo si la persona es signo Aries, caso contrario imprima no es signo Aries.
8. Ingresar por teclado el nombre, la edad y el sexo de cualquier persona e imprima, solo si la persona es de sexo masculino y mayor de edad, el nombre de la persona.
9. Hacer un Programa que borre la pantalla al pulsar 1.
10. Una distribuidora de motocicletas tiene una promoción de fin de año que consiste en lo siguiente. Las motos marca Honda tienen un descuento del 5%, las marcas Yamaha del 8% y las Suzuki del 10%, las otras marcas 2%.

Libro Práctico de Programación en C

11. Dada una nota de un examen mediante un código escribir el literal que le corresponde a la nota:
A - Excelente
B - Notable
C - Aprobado
D y F – Reprobado.
12. Seleccionar un tipo de vehículo e indicar el peaje a pagar según un valor numérico
1 - turismo, peaje = \$500.
2 - autobús, peaje = \$3000.
3 - motocicleta, peaje = \$300.
Caso contrario - Vehículo no autorizado.
13. Cambiar un número entero con el mismo valor pero en romanos.
14. Mostrar los meses del año, pidiéndole al usuario un número entre (1-12), y mostrar el mes al que corresponde.
15. Hacer un programa que simule un cajero automático con un saldo inicial de 1000 Dólares.
16. Hace un menú que considere las siguientes opciones:
Caso 1: Cubo de un numero
Caso 2: Numero par o impar
Case 3: salir.

Nivel 3: Estructuras de Control – Bucles o Ciclos:

1. Suma de los n primeros números.
2. Digite un número, si el número supera a 10, multiplique los 10 primeros números, sino, súmelos.
3. Múltiplos de 3 desde 1 hasta n.
4. Múltiplos de 5 desde 1 hasta n.
5. Sumar 1-2+3-4...
6. Sumar pares desde n hasta m.
7. Suma de los 10 primeros números pares.
8. Hacer un árbol con '*' del tipo: Ejemplo:

Digite el número de filas: 5

```
*  
**  
***  
****  
*****
```

9. Determinar si un Número es Primo o no.
10. Factorial de un número.
11. Suma de factoriales
12. Serie Fibonacci: 1 1 2 3 5 8 13...

13. Hacer un programa que imprima la suma de todos los números pares que hay desde 1 hasta n, y diga cuantos números hay.
14. Hacer un bucle do...while para imprimir las letras minúsculas del alfabeto.

Nivel 4: Funciones y Procedimientos:

1. Determinar si un número es par o no.
2. Hacer un programa que realice la media aritmética de 2 números.
3. Hacer un programa que muestre un menú con las opciones sumar, restar, multiplicar y dividir, el programa solicitará una opción y realizará la tarea elegida, se debe usar procedimientos.
4. Hacer un programa que pida por pantalla un número del 1 al 10 y mediante un procedimiento muestre por pantalla el número escrito en letras.
5. Hacer un programa que pida por pantalla una temperatura en grados Celsius, muestre un menú para convertirlos a Fahrenheit o Kelvin y muestre el equivalente por pantalla, utilizar funciones.
Celsius a Fahrenheit:
$$F = (9 \times C) / 5 + 32;$$
Celsius a Kelvin:
$$K = C + 273.15;$$
6. Hacer un programa que muestre una tabla de multiplicar hasta el 20 de un número cualquiera por pantalla, el número se pedirá en el programa principal. Usar procedimiento.
7. Hacer un programa que muestre 3 números ordenados de ascendentemente, utilizar un procedimiento para cada operación.
8. Hacer un programa que muestre 3 números ordenados descendientemente, utilizar un procedimiento para cada operación.

Recursividad:

1. Serie Fibonacci con Recursividad: 1.1.2.3.5.8...
2. Consideremos una función que recibe un número n e imprime los números del n al 1. Con recursividad.
3. Pasar de número entero a número binario con Recursividad.
4. Invertir un número entero con Recursividad.

Nivel 5: Arrays (Listas y Tablas):

1. Copiar el contenido de un Array1 de 5 elementos hacia otro Array2.
2. Crear un programa que tenga un array de 100 números aleatorios entre 1 y 1000. Una vez creado organizarlo de tal manera que almacene los números pares en un array y los impares en otro.
3. Hacer 2 arrays de cadena de caracteres, pedir el nombre al usuario guardarlo dentro de un array y luego, copiar el contenido de ese array a otro.
4. Hacer 2 arrays de cadenas de caracteres, almacenar 2 nombres cada uno dentro de su respectiva variable, y luego intercambiar el contenido de los arrays.

5. Realice un programa que lea en un array el sexo de los N estudiantes del curso de Algoritmos y que determine cuantos hombres y cuantas mujeres se encuentran en el grupo, suponiendo que los datos son extraídos alumno por alumno.
6. Realice un programa que permita leer una cadena de caracteres y que devuelva el número de caracteres que tiene dicha cadena e imprimir dicha cadena al revés.
7. Hacer una matriz de tipo entera preguntándole al usuario el número de filas y el número de columnas, rellenar la matriz y luego mostrarla en pantalla.
8. Hacer una matriz de tipo entera de $2 * 2$, llenarla de números y luego copiar todo su contenido hacia otra matriz.
9. Sumar 2 matrices.
10. Sumar 2 matrices, pidiendo al usuario el número de filas y columnas y además los elementos de cada matriz, luego mostrar el resultado.
11. Hacer una lista de 10 números, luego pedirle al usuario que digite un número, y a través de una búsqueda secuencial determinar si el número existe en la lista o no, y además indicar en qué posición se encuentra.
12. Hacer un array pidiendo al usuario el número de elementos, y rellenar el array con números digitados por el teclado, luego pedir al usuario un número, y hacer una búsqueda secuencial para indicar si ese elemento existe o no, también entregar posición.
13. Hacer un array de 10 números desordenados, luego ordenarlos con el método burbuja, posteriormente pedir un dato a buscar y utilizar la búsqueda binaria para determinar si existe o no.
14. Hacer un array de 10 números desordenados, luego ordenarlos con el ordenamiento por selección, posteriormente pedir un dato a buscar y utilizar la búsqueda binaria para determinar si existe o no.

Nivel 6: Estructuras:

1. Hacer una estructura llamada corredor, en la cual se tendrán los siguientes miembros: Nombre, edad, sexo, club, pedir datos al usuario para un corredor, y asignarle una categoría de competición:
 - Juvenil ≤ 18 años
 - Señor ≤ 40 años
 - Veterano > 40 añosPosteriormente imprimir todos los datos del corredor, incluida su categoría de competición.
2. Hacer una estructura llamada alumno, en la cual se tendrán los siguientes miembros: Nombre, edad, promedio, pedir datos al usuario para 3 alumnos, comprobar cuál de los 3 tiene el mejor promedio y posteriormente imprimir los datos del alumno.
3. Realizar un programa en C que lea un arreglo de estructuras los datos de N empleados de la empresa y que imprima los datos del empleado con mayor y menor salario.

4. Hacer un arreglo de estructura llamada atleta para N atletas que contenga los siguientes miembros: nombre, país, numero_medallas. y devuelva los datos (Nombre, país) del atleta que ha ganado el mayor número de medallas.
5. Hacer 2 estructuras una llamada promedio que tendrá los siguientes miembros: nota1, nota2, nota3; y otra llamada alumno que tendrá los siguientes miembros: nombre, sexo, edad; hacer que la estructura promedio este anidada en la estructura alumno, luego pedir todos los datos para un alumno, luego calcular su promedio, y por ultimo imprimir todos sus datos incluidos el promedio.
6. Utilizar las 2 estructuras del problema 5, pero ahora pedir los datos para N alumnos, y calcular cuál de todos tiene el mejor promedio, e imprimir sus datos.

Nivel 7: Punteros (Apuntadores):

1. Hacer una variable de tipo int, otra de tipo float y por ultimo una de tipo char, almacenar datos en cada una de las variables, posteriormente indicar la posición de memoria donde se encuentran guardados los datos de cada variable. Con punteros.
2. Comprobar si un número es par o impar, y señalar la posición de memoria donde se está guardando el número. Con punteros.
3. Imprimir el alfabeto en mayúscula con Punteros.
4. Determinar si un número es primo o no; con punteros y además indicar en que posición de memoria se guardó el número.
5. Dado un vector de 10 elementos = {1, 2, 3, 4, 4, 7, 8, 9, 5, 4}, escribir un programa en C (haciendo uso de puntero) que muestre las direcciones de memoria de cada elemento del vector.
6. Rellenar un array de 10 números, posteriormente utilizando punteros indicar cuales son números pares y su posición en memoria.
7. Pedir su nombre al usuario y devolver el número de vocales que hay.
8. Pedir una cadena de caracteres (string) al usuario, e indicar cuantas veces aparece la vocal a, e, i, o, u; en la cadena de caracteres.

Nivel 8: Asignación Dinámica de Memoria:

1. Utilizar malloc para reservar memoria para un nombre (string).
2. Pedir una cadena de caracteres al usuario, posteriormente utilizar malloc para reservar memoria para esa cadena de caracteres (string).
3. Reservar memoria con malloc para un array de tipo double para 1000 números, y comprobar si el puntero es válido (si el puntero devuelve NULL, significa que no hay suficiente memoria).
4. Pedir al usuario el número de elementos para un array de números enteros, posteriormente comprobar con malloc si hay memoria suficiente (si el puntero es válido, es decir diferente de NULL), si la hay es entonces rellenar ese array con números aleatorios, si no la hay decir que hay un error en la asignación de memoria.

5. Reservar espacio en memoria dinámica para un nombre, y luego liberar la memoria utilizada con `free()`.
6. Pedir al usuario una cadena de caracteres, reservar memoria con `malloc()`, almacenarlo en la memoria dinámica, posteriormente utilizar `free()` para liberar el espacio de memoria dinámica utilizado.
7. Reservar memoria dinámica para una cadena de caracteres de 10 espacios, luego utilizar `realloc` para ampliar la memoria dinámica a 30 espacios.
8. Reservar memoria para un arreglo de 5 números, rellenarlo con números que digite el usuario, luego utilizar `realloc` para ampliar ese arreglo a 10 números y agregar 5 números más, por ultimo imprimir el arreglo resultante de 10 números.
9. Reservar memoria con `calloc` para una cadena de caracteres.
10. Hacer un puntero array de tipo `float`, pedir el usuario el número de elementos, luego reservar memoria dinámica con `calloc`, luego llenar el array, imprimir todos los elementos, y por ultimo liberar el espacio de memoria dinámica utilizado.

Nivel 9: Cadenas:

1. Utilizar las funciones `getchar()` y `putchar()` para leer e imprimir un string.
2. Leer una cadena de caracteres con `getchar()` y contabilizar cuantos espacios ocupa.
3. Leer un string con `getchar()` y contabilizar cuantas vocales tiene.
4. Leer un string con `getchar()` y contabilizar cuantas veces aparece la vocal a, e, i, o, u, e imprimir el contador de cada una.
5. Copiar la cadena de caracteres "Hola mundo" de un array a otro. Luego cambiar la palabra "mundo" por "y bienvenido al lenguaje C".
6. Pedir al usuario que digite una cadena, luego copiar su contenido hacia otro array de caracteres. y por ultimo modificar su mensaje poniendo a partir de la posición 10 "Buena broma".
7. Utilizar la función `strcat()` para añadir un texto a un string.
8. Pedir una cadena al usuario, posteriormente utilizar la función `strcat()` para añadir la frase "Que tengas un buen día", al final de la cadena.
9. Utilizar la función `strcmp()` para comparar dos palabras.
10. Pedir 2 nombres al usuario y comprar si son iguales o no.
11. Pedir 2 palabras al usuario y ordenarlas alfabéticamente.
12. Pedir n palabras al usuario y posteriormente ordenarlas alfabéticamente. Utilizar método de ordenamiento (Burbuja, inserción, selección).
13. Invertir una cadena con la función `strrev()`.
14. Pedir una cadena al usuario y luego invertirla. diciendo además cuantos caracteres tiene.
15. Convertir dos cadena de minúsculas a MAYUSCULAS con la función `strupr()`. Compararlas, y decir si son iguales.
16. Pedir al usuario su nombre, apellido, y ciudad en la que vive, posteriormente convertir todo a MAYUSCULAS e imprimir los datos.(Utilizar una estructura).

17. Pedir su nombre y apellidos al usuario (en mayúscula), posteriormente utilizar la función `strlwr()` para convertir las MAYUSCULAS en minúsculas e imprimir los datos.
18. Pedir el nombre de 3 familiares al usuario (en MAYUSCULAS), y posteriormente convertir los datos a minúsculas.
19. Determinar si una palabra es capicúa (palíndroma) o no.
20. Pedir al usuario que digite una palabra. luego mostrar la palabra invertida y comprobar si es capicúa.
21. Convertir una cadena (por ejemplo: "1234"), a un número entero. Con la función `atoi()`.
22. Pedir dos cadenas numéricas al usuario. Luego Convertirlas a números enteros. Sumar los valores y entregar el resultado.
23. Utilizar la función `atof()` para pasar una cadena (por ejemplo: "123.45"), a número flotante.
24. Pedir dos cadenas(con numero flotante) al usuario, posteriormente multiplicar dichos números y mostrar su resultado.

Nivel 10: Entradas y Salidas por Archivos:

1. Utilizar la función `fopen()`, para determinar si existe un archivo de texto (.txt) o no.
2. Crear un archivo de texto (.txt) en la ubicación que elijas, posteriormente crear un programa en C, que localice tu archivo de texto creado.
3. Hacer un programa en C que cree un archivo de texto (.txt).
4. Hacer un programa en C que cree un archivo de texto (.txt) llamado "hola mundo", crear el archivo en la carpeta que elija.
5. Abrir 2 archivos de texto (.txt), después cerrar cada uno de ellos con `fclose()`.
6. Hacer un programa en C, que cree 2 archivos de texto (.txt), y luego utilizar la función `fclose()` para cerrarlos.
7. Utilizar la función `fputc()` para introducir caracteres en un archivo de texto (.txt).
8. Crear un archivo de texto (.txt) con el nombre de "hola mundo" en la carpeta que elijas, y dentro del archivo copiar "Hola y Bienvenidos al Lenguaje de Programación C".
9. Tomar el archivo de texto (.txt) creado en el problema 7, y con la función `fgetc()` leer todo su contenido.
10. Tomar el archivo de texto creado en el problema 8. y leer todo su contenido.
11. Agregar texto a un archivo txt.
12. Crear un archivo de texto, ponerle el texto que desees, y luego crear un programa en C para agregarle más texto al archivo.
13. Crear un archivo de texto (.txt), donde guardar los emails de amigos.
14. Crear un programa en C, que pueda seguir agregando contactos de email, hacia el archivo que creamos en el problema 13.

15. Hacer un programa en C. Para guardar direcciones de correo que muestre un menú con las siguientes opciones:

1. Crear (nombre, apellidos, email)
2. Agregar más contactos (nombre, apellidos, email)
3. Visualizar contactos existentes

16. Hacer un programa en C. Para guardar los números de celular, que muestre un menú con las siguientes opciones:

1. Crear (nombre, apellidos, celular)
2. Agregar más contactos (nombre, apellidos, celular)
3. Visualizar contactos existentes

Nivel 11: Listas Simplemente Enlazadas:

1. Crea una lista simplemente enlazada de 3 números flotantes, pidiéndole al usuario dichos números.
2. Crea una lista simplemente enlazada de 3 caracteres, pidiéndole al usuario dichos caracteres.