Ejercicios programacion C

# Programa 1

Escribe un programa en lenguaje C que muestre en la consola el mensaje "Mi primer programa.", espere a que el usuario presione la tecla Intro y, posteriormente, finalice su ejecución.

#include <stdio.h>

#include <conio.h>

int main()

{

printf("Mi primer programa.");

printf("pulsa una tecla para acabar ...");

getch();

}

# Programa 2

Escribe un programa en C que declare diferentes variables para representar datos básicos. Declara variables de tipo char, int y float para representar un caracter, un número entero y un número decimal, respectivamente. Luego, en una misma línea, declara otras variables para cada tipo de dato, para mostrar que es posible declarar múltiples variables del mismo tipo en una sola instrucción.

#include <stdio.h>

#include <stdlib.h>

int main(void)

{

int i = 0, j = 4, h = 2;

char a[] = "abcde";

float f = 3.56;

printf("%f, %d, %d, %c\n", f, i, h, a[3]);

return 0;

}

# Programa 3

Escribe un programa en C que utilice la directiva de preprocesador #define para asignar un valor constante (en este caso, el número 8) a la macro NUMERO. El programa deberá mostrar en la consola el valor asignado a NUMERO utilizando la instrucción printf. Finalmente, esperará a que el usuario presione Intro antes de finalizar.

# Programa 4

Escribe un programa en C que declare dos variables de tipo float para representar coordenadas en el plano (X e Y), asignándoles los valores 38.35646 y -0.6540 respectivamente. El programa deberá mostrar ambos valores en la consola utilizando printf y esperar a que el usuario presione Intro antes de finalizar.

# Programa 5

Escribe un programa en C que declare e inicialice varias variables enteras (u, v, w, x, y, z) para demostrar el uso de diferentes operadores aritméticos básicos: suma, resta, multiplicación, división y módulo. El programa deberá mostrar en la consola los resultados de cada operación, junto con la versión negativa de una de las variables, para ilustrar cómo representar un número en negativo. Al finalizar, esperará a que el usuario presione Intro antes de cerrar.

# Programa 6

Escribe un programa en C que demuestre el uso de operadores de asignación compuestos (+= y -=). El programa debe declarar dos variables enteras (x e y), asignándoles inicialmente los valores 8 y 2, respectivamente. Luego, modifica el valor de x sumándole 2 mediante +=, y modifica el valor de y restándole 2 mediante -=. Finalmente, el programa mostrará en pantalla los valores resultantes de x e y y esperará a que el usuario presione Intro antes de finalizar.

# Programa 7

Escribe un programa en C que demuestre el uso de los operadores de incremento (++) y decremento (--) en variables enteras. El programa inicializa dos variables (x = 8 e y = 2), aplica primero los operadores de incremento y decremento en su versión postfija para aumentar y disminuir su valor en una unidad, e imprime los resultados. Luego, aplica la versión prefija de estos operadores para aumentar y disminuir nuevamente los valores antes de utilizarlos, e imprime los resultados finales. El programa concluye esperando a que el usuario presione Intro antes de finalizar.

# Programa 8

Escribe un programa en C que solicite al usuario ingresar su edad, el año actual y el año de nacimiento utilizando scanf. El programa deberá guardar estos datos en variables de tipo entero y, a continuación, mostrarlos en la consola para confirmar que fueron registrados correctamente. El programa concluye esperando a que el usuario presione Intro antes de finalizar.

# Programa 9

Escribe un programa en C que demuestre el uso de punteros para asignar y acceder a valores de variables. El programa declarará variables para representar lápices, libros y calculadoras junto con sus respectivos punteros. Luego, asignará a los punteros las direcciones de las variables, les asignará valores a través de la desreferenciación (\*), e imprimirá tanto los valores asignados como la dirección de memoria correspondiente para cada caso. Finalmente, esperará a que el usuario presione Intro antes de finalizar.

# Programa 10

Escribe un programa en C que utilice la función putchar() para mostrar en la consola un caracter literal y otro caracter guardado en una variable. El programa debe declarar una variable de tipo char (con valor 'K') e imprimir primero la letra 'H' y luego la variable letra. Al finalizar, mostrará un mensaje para que el usuario presione Intro antes de cerrar.

# Programa 11

Escribe un programa en C que lea un caracter desde teclado utilizando la función getchar() y lo muestre inmediatamente en la consola mediante putchar(). Después, debe esperar a que el usuario presione Intro antes de finalizar la ejecución.

# Programa 12

Escribe un programa en C que utilice la función getch() para capturar un caracter del teclado sin necesidad de presionar Intro, y lo muestre inmediatamente en la consola mediante putchar(). Después, el programa deberá esperar a que el usuario presione cualquier tecla para finalizar.

# Programa 13

Escribe un programa en C que utilice la función getche() para capturar un caracter desde teclado y mostrarlo inmediatamente en la consola al momento de presionarse. Después, el programa esperará a que el usuario presione cualquier tecla para finalizar la ejecución.

# Programa 14

Escribe un programa en C que utilice la función printf() para mostrar en pantalla el mensaje "Hola mundo!". Después, el programa deberá esperar a que el usuario presione cualquier tecla antes de finalizar la ejecución.

# Programa 15

Escribe un programa en C que demuestre el uso de diferentes caracteres especiales en la consola. El programa debe mostrar:

* Texto con tabulación (\t),
* Texto en una nueva línea (\n),
* Texto entre comillas dobles (\"), y
* Texto entre comillas simples (\').

Al finalizar, el programa esperará a que el usuario presione cualquier tecla para cerrar.

# Programa 16

Escribe un programa en C que utilice la función printf() para mostrar el valor de una variable entera (x) en la consola utilizando un especificador de formato (%d). El programa asignará a x el valor 8 y lo mostrará junto al texto "X =". Al finalizar, esperará a que el usuario presione cualquier tecla antes de cerrar.

# Programa 17

Escribe un programa en C que demuestre el uso de diferentes especificadores de formato para representar variables de distintos tipos en la consola. El programa deberá declarar e inicializar un entero (*int*), un real (*float*) y un caracter (*char*), y mostrar:

* El caracter utilizando %c.
* El entero utilizando %d.
* El real utilizando %f y con dos dígitos de precisión (%.2f).
* El entero y el real alineados en un campo de diez dígitos (%10d y %10f).
* La dirección de memoria de la variable caracter (%p).

Al finalizar, esperará a que el usuario presione cualquier tecla para cerrar.

# Programa 18

Escribe un programa en C que demuestre diferentes representaciones de datos utilizando especificadores de formato en la consola. El programa deberá declarar una variable de tipo caracter (*char*) y dos variables de tipo entero (*int*), asignándoles los valores *'H'*, *72* y *8* respectivamente. Luego, mostrará en la consola:

* El valor ASCII numérico correspondiente al caracter H.
* El caracter correspondiente al valor ASCII 72.
* El entero 8 en formato octal.
* El entero 8 en formato hexadecimal.
* El entero 8 mostrado como un número de punto flotante con dos dígitos de precisión (%.2f).

Al finalizar, el programa esperará a que el usuario presione cualquier tecla para cerrar.

# Programa 19

Escribe un programa en C que demuestre la impresión de un mismo valor entero en diferentes formatos dentro de una misma línea. El programa declarará una variable entera (*numero*) con el valor *72* y la mostrará en la consola como:

* Entero decimal (%d)
* Caracter correspondiente al valor ASCII (%c)
* Entero en formato octal (%o)
* Entero en formato hexadecimal (%x)
* Número de punto flotante con dos dígitos de precisión (%.2f)

Al finalizar, el programa esperará a que el usuario presione cualquier tecla para cerrar.

# Programa 20

Escribe un programa en C que solicite al usuario ingresar un valor entero para asignarlo a la variable *x* utilizando la función *scanf()*. El programa mostrará el valor registrado en pantalla junto al texto "x = " para confirmar que la operación de lectura fue realizada correctamente. Al finalizar, esperará a que el usuario presione cualquier tecla para cerrar.

# Programa 21

*Escribe un programa en C que solicite al usuario ingresar un número entero y utilice estructuras condicionales (if) para evaluar si el valor es positivo y si tiene un solo dígito (entre 0 y 9). El programa mostrará en la consola mensajes que indiquen si el número es positivo y si tiene un dígito. Al finalizar, esperará a que el usuario presione cualquier tecla para cerrar.*

# Programa 22

*Escribe un programa en C que solicite al usuario ingresar un número entero y utilice estructuras condicionales if...else para evaluar:*

* *Si el número es positivo o negativo.*
* *Si el número tiene un solo dígito (de 0 a 9) o más de uno.*

*El programa mostrará en la consola mensajes acordes al resultado de cada evaluación y esperará a que el usuario presione cualquier tecla para finalizar.*

# Programa 23

*Escribe un programa en C que solicite al usuario ingresar una vocal y utilice la estructura de control switch para evaluar la entrada. El programa mostrará un mensaje específico para cada vocal (a, e, i, o, u) e indicará al usuario cuando no haya ingresado una vocal válida. Al finalizar, esperará a que el usuario presione cualquier tecla para cerrar.*

# Programa 24

*Escribe un programa en C que utilice bucles while para mostrar y evaluar datos ingresados por el usuario. El programa deberá:*

1. *Imprimir todos los números del 0 al 10 en la consola.*
2. *Solicitar al usuario ingresar un número entre 1 y 10, repitiendo la solicitud hasta que el usuario ingrese el número 8.*

*Una vez alcanzada esta condición, mostrará un mensaje para que el usuario presione cualquier tecla para finalizar.*

# Programa 25

*Escribe un programa en C que utilice un bucle do...while para mostrar al usuario un menú con dos opciones:*

1. Continuar
2. Salir

*Mientras el usuario seleccione la opción 1 (Continuar), el menú se repetirá. Cuando el usuario ingrese 2 (Salir), el programa finalizará mostrando un mensaje para que presione cualquier tecla antes de cerrar.*

# Programa 26

*Escribe un programa en C que utilice un bucle for para mostrar en pantalla todos los números del 0 al 10, separados por espacios. Al finalizar, mostrará un mensaje para que el usuario presione cualquier tecla antes de cerrar.*

# Programa 27

*Escribe un programa en C que utilice sentencias anidadas para mostrar todos los números pares del 0 al 10. El programa empleará un bucle while para recorrer los números del 0 al 10 y una condición if para verificar si cada número es par antes de mostrarlo en la consola. Al finalizar, esperará a que el usuario presione cualquier tecla para cerrar.*

# Programa 28

*Escribe un programa en C que permita al usuario ingresar cinco letras, una por una, y almacene cada una en un array de caracteres. El programa deberá contar cuántas de esas letras son 'a' o 'A' e informar al usuario cuántas coincidencias hubo. Al finalizar, esperará a que el usuario presione cualquier tecla para cerrar.*

# Programa 29

*Escribe un programa en C que permita al usuario llenar una matriz de 5 filas y 3 columnas con valores enteros proporcionados por teclado. El programa primero pedirá al usuario que ingrese un valor para cada posición de la matriz, y posteriormente mostrará todos los datos organizados en formato de matriz. Al finalizar, esperará a que el usuario presione cualquier tecla para cerrar.*

# Programa 30

*Escribe un programa en C que solicite al usuario ingresar una palabra (cadena de caracteres) y la almacene en un array de char. Luego, mostrará la palabra completa en la consola para confirmar la correcta captura de datos. Al finalizar, esperará a que el usuario presione cualquier tecla para cerrar.*

# Programa 31

*Escribe un programa en C que permita al usuario ingresar una palabra y la almacene en un array de char. Luego, utilizando la función strcpy(), copiará esta palabra a otro array y mostrará la copia en la consola para verificar que la operación se realizó correctamente.*

# Programa 32

*Escribe un programa en C que utilice las funciones strcpy() y strcat() para trabajar con cadenas de caracteres. El programa copiará primero los textos "Horc" y "kun" a dos arreglos de char y luego concatenará ambos para formar "Horckun". Finalmente, mostrará la cadena resultante en la consola y esperará a que el usuario presione cualquier tecla para cerrar.*

# Programa 33

*Escribe un programa en C que solicite al usuario ingresar una palabra y utilice la función strlen() para determinar la cantidad de letras que ésta contiene. El programa mostrará en la consola la palabra junto con la cantidad de caracteres que la componen, y esperará a que el usuario presione cualquier tecla para finalizar.*

# Programa 34

*Escribe un programa en C que permita al usuario ingresar dos palabras y utilice la función strcmp() para compararlas. El programa determinará si las palabras son iguales o diferentes e imprimirá un mensaje correspondiente en la consola. Al finalizar, simplemente concluirá la ejecución.*

# Programa 35

*Escribe un programa en C que permita al usuario ingresar una frase y una palabra, y utilice la función strstr() para determinar si la palabra existe dentro de la frase. El programa mostrará un mensaje indicando si la palabra fue encontrada o no. Al finalizar, simplemente concluirá la ejecución.*

# Programa 36

*Escribe un programa en C que permita al usuario ingresar una palabra en mayúsculas y utilice la función strlwr() para convertirla a minúsculas. El programa mostrará la palabra resultante en la consola y finalizará la ejecución.*

# Programa 37

*Escribe un programa en C que permita al usuario ingresar una palabra en minúsculas y utilice la función strupr() para convertirla a mayúsculas. El programa mostrará la palabra convertida en la consola y finalizará su ejecución.*

# Programa 38

*Escribe un programa en C que permita al usuario ingresar una serie de palabras separadas por comas y utilice la función strtok() para dividir la cadena en cada palabra. El programa mostrará cada palabra en una línea independiente en la consola y finalizará al procesarse todas.*

# Programa 39

*Escribe un programa en C que permita al usuario ingresar dos números como cadenas de texto, utilice la función atoi() para convertir dichas cadenas a enteros, y posteriormente calcule e imprima la suma de ambos en la consola.*

# Programa 40

*Escribe un programa en C que utilice una estructura (struct) para guardar la información de una persona, incluyendo su nombre y edad. El programa solicitará al usuario que ingrese estos datos por teclado y, a continuación, los mostrará en la consola para confirmar que fueron registrados correctamente.*

# Programa 41

*Escribe un programa en C que demuestre la estructura y uso de una función definida por el usuario. El programa contará con una función secundaria llamada funcion() que imprime un mensaje enmarcado con caracteres especiales ("Hello World!"). La función principal (main()) llamará a esta función para mostrar el mensaje en consola y esperará a que el usuario presione una tecla para finalizar.*

# Programa 42

*Escribe un programa en C que demuestre el paso de parámetros por valor a través de una función definida por el usuario. El programa solicitará al usuario un número entero y lo pasará a una segunda función llamada funcion(). Esta última mostrará el valor recibido en la consola. Finalmente, el programa esperará a que el usuario presione cualquier tecla para cerrar.*

# Programa 43

*Escribe un programa en C que permita al usuario ingresar dos números enteros y utilice una función definida por el usuario para obtener la suma de ambos. El programa deberá pedir ambos números, llamar a la función suma() para obtener el resultado y mostrarlo en la consola. Finalmente, esperará a que el usuario presione cualquier tecla para cerrar.*

# Programa 44

*Escribe un programa en C que demuestre el paso de parámetros por referencia. El programa solicitará al usuario un valor para la variable x, mostrará este valor en consola y, posteriormente, llamará a una función llamada cambiar() para incrementarlo en 1. Al finalizar, el programa mostrará el nuevo valor de x para verificar que la modificación realizada por referencia se reflejó correctamente. Finalmente, esperará a que el usuario presione una tecla para cerrar.*

# Programa 45

*Escribe un programa en C que demuestre la diferencia entre variables locales y globales. El programa declarará una variable local en la función principal (main()), le asignará un valor y la mostrará en la consola. Luego llamará a una función secundaria (funcion()), que asignará un valor a la variable global definida al inicio del programa. Finalmente, main() mostrará en pantalla el valor de la variable global para verificar que esta fue modificada correctamente. Al finalizar, esperará a que el usuario presione una tecla para cerrar.*

# Programa 46

*Escribe un programa en C que calcule el factorial de un número entero utilizando una función definida por el usuario. El programa pedirá al usuario que ingrese un número, inicializará la variable de resultado (f) a 1 y llamará a la función factorial() para realizar el cálculo. Dentro de esta, mostrará de manera progresiva cada paso de la operación y actualizará el valor de f. Finalmente, el programa mostrará en consola el factorial calculado y esperará a que el usuario presione cualquier tecla para finalizar.*

# Programa 47

*Escribe un programa en C que demuestre la apertura de un archivo en modo lectura de texto (rt). El programa intentará abrir un archivo llamado archivo.txt y verificará si la operación tuvo éxito. Si el archivo no se puede abrir, mostrará un mensaje de error en la consola. Al finalizar, esperará a que el usuario presione cualquier tecla para cerrar.*

# Programa 48

*Escribe un programa en C que abra un archivo de texto llamado archivo.txt en modo de lectura (rt), compruebe si la apertura fue correcta e informe al usuario en caso de error. Si el archivo se abre correctamente, el programa lo cerrará utilizando la función fclose(). Al finalizar, mostrará un mensaje para que el usuario presione cualquier tecla antes de cerrar.*

# Programa 49

*Escribe un programa en C que abra un archivo llamado archivo.txt para escritura (modo texto), verifique si la apertura fue exitosa y utilice la función fputc() para insertar un carácter ('H') en el archivo. Después, cerrará el archivo correctamente y mostrará un mensaje para que el usuario presione cualquier tecla antes de finalizar.*

# Programa 50

*Escribe un programa en C que abra un archivo llamado archivo.txt en modo escritura ("w"), verifique si la apertura fue exitosa y utilice las funciones fputc() y fputs() para insertar un carácter ('H') y una cadena de texto ("Hola mundo!") en el archivo, respectivamente. Después, cerrará el archivo correctamente y mostrará un mensaje para que el usuario presione cualquier tecla antes de finalizar.*

# Programa 51

*Escribe un programa en C que abra un archivo llamado archivo.txt en modo escritura de texto ("wt"), verifique si la apertura fue exitosa y utilice las funciones fputc(), fputs() y fwrite() para escribir un carácter, una cadena de texto y un bloque de datos respectivamente en el archivo. En este caso, el programa escribirá el carácter 'H' dos veces (una con fputc y otra con fwrite) y la cadena "Hola mundo!" en medio. Finalmente, cerrará el archivo y mostrará un mensaje para que el usuario presione cualquier tecla antes de finalizar.*

# Programa 52

*Escribe un programa en C que abra un archivo llamado archivo.txt en modo lectura de texto ("rt"), verifique si la apertura fue exitosa y utilice la función fgetc() para leer un carácter del archivo. El programa mostrará el carácter leído en la consola y cerrará el archivo correctamente. Finalmente, esperará a que el usuario presione cualquier tecla para finalizar.*

# Programa 53

*Escribe un programa en C que abra un archivo llamado archivo.txt en modo lectura de texto ("rt"), compruebe si el archivo se abrió correctamente y lea su contenido carácter por carácter utilizando fgetc(). El programa deberá usar la función feof() para detectar el final del archivo y mostrar en la consola todo su contenido hasta ese momento. Finalmente, cerrará el archivo y esperará a que el usuario presione cualquier tecla para finalizar.*

# Programa 54

*Escribe un programa en C que abra un archivo llamado archivo.txt en modo lectura de texto ("rt"), verifique si se abrió correctamente y lea su contenido línea por línea utilizando la función fgets(). El programa mostrará cada línea leída en la consola hasta que alcance el final del archivo. Finalmente, cerrará el archivo y esperará a que el usuario presione cualquier tecla para finalizar.*

# Programa 55

*Escribe un programa en C que abra un archivo llamado archivo.txt en modo lectura de texto ("rt") y utilice la función fseek() para posicionarse en una ubicación específica dentro del archivo, basada en el tamaño de una estructura ficha. Luego, usará fread() para leer datos desde esa posición y mostrará el nombre almacenado en la estructura en la consola. Finalmente, cerrará el archivo y esperará a que el usuario presione cualquier tecla para finalizar.*

# Programa 56

*Escribe un programa en C que utilice asignación dinámica de memoria para manejar una estructura ficha que contiene un nombre y una edad. El programa debe reservar memoria con malloc(), verificar que la reserva fue exitosa, solicitar al usuario que ingrese el nombre y la edad, mostrar estos datos en la consola y finalmente liberar la memoria asignada con free(). Al finalizar, esperará a que el usuario presione cualquier tecla para cerrar.*

# Programa 57

*Escribe un programa en C que implemente el método de la burbuja para ordenar un arreglo de enteros de tamaño 4. El programa debe ordenar el arreglo vector de manera ascendente y luego mostrar sus elementos ordenados en la consola. Al finalizar, esperará a que el usuario presione cualquier tecla para cerrar.*

# Programa 58

*Escribe un programa en C que implemente el método de selección directa para ordenar un arreglo de 5 enteros de forma ascendente. El programa debe ordenar el arreglo v y luego mostrar sus elementos ordenados en la consola. Al finalizar, esperará a que el usuario presione cualquier tecla para cerrar.*

**Nota:** En el código, el límite del segundo bucle for debería ser j < 5 para recorrer correctamente todos los elementos y evitar saltarse la última posición.

# Programa 59

*Escribe un programa en C que implemente el método de búsqueda secuencial para encontrar un número en un arreglo de 5 enteros. El programa solicitará al usuario que ingrese un número, recorrerá el arreglo para determinar si el número está presente y mostrará un mensaje indicando si el número fue encontrado o no. Finalmente, esperará a que el usuario presione cualquier tecla para cerrar.*

# Programa 60

*Escribe un programa en C que solicite al usuario ingresar su edad y valide que el valor ingresado sea correcto, es decir, que esté entre 0 y 120. Si el usuario introduce un valor fuera de este rango, el programa mostrará un mensaje de error y volverá a solicitar la edad hasta que se ingrese un valor válido. Finalmente, el programa esperará a que el usuario presione cualquier tecla para finalizar.*

# Programa 61

*Escribe un programa en C que cuente el número de palabras en una cadena de texto almacenada en un arreglo de caracteres. Para ello, el programa contará los espacios en blanco que separan las palabras dentro de la cadena y mostrará la frase, el total de caracteres y la cantidad de espacios encontrados en la consola. Finalmente, esperará a que el usuario presione cualquier tecla para finalizar.*

# Programa 62

*Escribe un programa en C que elimine todos los espacios de una cadena de caracteres dada y muestre la cadena resultante sin espacios en la consola. El programa debe procesar una cadena predefinida que contiene espacios innecesarios y mostrar el texto limpio de espacios. Finalmente, esperará a que el usuario presione cualquier tecla para finalizar.*

# Programa 63

*Escribe un programa en C que solicite al usuario ingresar una fecha compuesta por día, mes y año. El programa debe verificar que el día esté entre 1 y 31, el mes entre 1 y 12, y el año entre 1919 y 2019. Si la fecha es válida, la mostrará en formato dd/mm/aaaa; en caso contrario, indicará que la fecha es incorrecta. Finalmente, esperará a que el usuario presione cualquier tecla para finalizar.*

# Programa 64

*Escribe un programa en C que solicite al usuario ingresar un año y determine si es bisiesto. El programa debe validar que el año esté en un rango razonable (0 a 2100) y verificar las condiciones correctas para un año bisiesto: divisible entre 400, o divisible entre 4 pero no entre 100. Mostrará un mensaje indicando si el año es bisiesto o no, o si el valor ingresado es incorrecto. Finalmente, esperará a que el usuario presione cualquier tecla para finalizar.*

# Programa 65

*Escribe un programa en C que permita validar un DNI español. El programa solicitará al usuario que ingrese el número y la letra del DNI, y verificará que la letra sea correcta según el número proporcionado utilizando una función que calcula la letra correspondiente a partir del módulo 23 del número. El programa repetirá la solicitud hasta que la letra ingresada coincida con la calculada, mostrando mensajes apropiados en cada caso. Finalmente, esperará a que el usuario presione cualquier tecla para finalizar.*