**1. Hola Mundo**

* **Descripción**: El clásico "Hola, Mundo!" para entender la estructura básica de un programa en C.
* **Código**:

c

Copy

#include <stdio.h> // Incluye la biblioteca estándar de entrada/salida

int main() { // Función principal, punto de entrada del programa

printf("Hola, Mundo!\n"); // Imprime "Hola, Mundo!" en la pantalla

return 0; // Indica que el programa terminó correctamente

}

* **Explicación**:
  + #include <stdio.h>: Incluye la biblioteca estándar de entrada/salida, necesaria para usar printf.
  + int main(): La función principal donde comienza la ejecución del programa.
  + printf: Función para imprimir texto en la consola.
  + \n: Carácter de nueva línea.
  + return 0: Indica que el programa terminó sin errores.

**2. Suma de dos números**

* **Descripción**: Pide al usuario dos números y muestra la suma.
* **Código**:

c

Copy

#include <stdio.h>

int main() {

int num1, num2, suma; // Declara variables para almacenar los números y la suma

printf("Introduce el primer número: "); // Solicita el primer número

scanf("%d", &num1); // Lee el primer número y lo almacena en num1

printf("Introduce el segundo número: "); // Solicita el segundo número

scanf("%d", &num2); // Lee el segundo número y lo almacena en num2

suma = num1 + num2; // Calcula la suma

printf("La suma es: %d\n", suma); // Muestra el resultado

return 0;

}

* **Explicación**:
  + int num1, num2, suma;: Declara variables de tipo entero.
  + scanf("%d", &num1);: Lee un número entero y lo almacena en num1.
  + &num1: El operador & obtiene la dirección de memoria de num1 para que scanf pueda almacenar el valor.
  + %d: Especificador de formato para enteros.

**3. Factorial de un número**

* **Descripción**: Calcula el factorial de un número ingresado por el usuario.
* **Código**:

c

Copy

#include <stdio.h>

int main() {

int n, factorial = 1; // Declara variables para el número y el factorial

printf("Introduce un número: "); // Solicita el número

scanf("%d", &n); // Lee el número y lo almacena en n

for (int i = 1; i <= n; i++) { // Bucle para calcular el factorial

factorial \*= i; // Multiplica factorial por i en cada iteración

}

printf("El factorial de %d es: %d\n", n, factorial); // Muestra el resultado

return 0;

}

* **Explicación**:
  + for (int i = 1; i <= n; i++): Bucle que se ejecuta desde i = 1 hasta i = n.
  + factorial \*= i;: Multiplica factorial por i y almacena el resultado en factorial.
  + %d: Especificador de formato para enteros.

**4. Verificar si un número es primo**

* **Descripción**: Verifica si un número ingresado por el usuario es primo.
* **Código**:

c

Copy

#include <stdio.h>

int main() {

int n, es\_primo = 1; // Declara variables para el número y un indicador de primalidad

printf("Introduce un número: "); // Solicita el número

scanf("%d", &n); // Lee el número y lo almacena en n

for (int i = 2; i <= n / 2; i++) { // Bucle para verificar divisores

if (n % i == 0) { // Si n es divisible por i

es\_primo = 0; // No es primo

break; // Sale del bucle

}

}

if (es\_primo) { // Si es\_primo es 1

printf("%d es un número primo.\n", n);

} else { // Si es\_primo es 0

printf("%d no es un número primo.\n", n);

}

return 0;

}

* **Explicación**:
  + es\_primo = 1;: Suponemos que el número es primo inicialmente.
  + n % i == 0: Verifica si n es divisible por i.
  + break;: Sale del bucle si se encuentra un divisor.

**5. Invertir una cadena**

* **Descripción**: Pide al usuario una cadena y muéstrala invertida.
* **Código**:

c

Copy

#include <stdio.h>

#include <string.h> // Incluye la biblioteca para manejar cadenas

int main() {

char cadena[100], temp; // Declara un arreglo para la cadena y una variable temporal

int inicio, fin;

printf("Introduce una cadena: "); // Solicita la cadena

scanf("%s", cadena); // Lee la cadena y la almacena en el arreglo

inicio = 0;

fin = strlen(cadena) - 1; // Obtiene la longitud de la cadena y resta 1 para el índice

while (inicio < fin) { // Bucle para invertir la cadena

temp = cadena[inicio];

cadena[inicio] = cadena[fin];

cadena[fin] = temp;

inicio++;

fin--;

}

printf("Cadena invertida: %s\n", cadena); // Muestra la cadena invertida

return 0;

}

* **Explicación**:
  + char cadena[100];: Declara un arreglo de caracteres para almacenar la cadena.
  + strlen(cadena): Devuelve la longitud de la cadena.
  + while (inicio < fin): Intercambia los caracteres desde el inicio hasta el final.

**6. Matriz transpuesta**

* **Descripción**: Pide al usuario una matriz y calcula su transpuesta.
* **Código**:

c

Copy

#include <stdio.h>

int main() {

int matriz[3][3], transpuesta[3][3]; // Declara matrices 3x3

printf("Introduce los elementos de la matriz 3x3:\n");

for (int i = 0; i < 3; i++) { // Bucle para leer la matriz

for (int j = 0; j < 3; j++) {

scanf("%d", &matriz[i][j]);

}

}

for (int i = 0; i < 3; i++) { // Bucle para calcular la transpuesta

for (int j = 0; j < 3; j++) {

transpuesta[j][i] = matriz[i][j];

}

}

printf("Matriz transpuesta:\n");

for (int i = 0; i < 3; i++) { // Bucle para mostrar la transpuesta

for (int j = 0; j < 3; j++) {

printf("%d ", transpuesta[i][j]);

}

printf("\n");

}

return 0;

}

* **Explicación**:
  + matriz[3][3]: Declara una matriz de 3x3.
  + transpuesta[j][i] = matriz[i][j];: Intercambia filas por columnas.

**Conclusión:**

Estos ejercicios cubren conceptos fundamentales de C, como variables, bucles, condicionales, arreglos y funciones. Practica cada uno de ellos en **Visual Studio Code** y asegúrate de entender cada línea de código. ¡Buena suerte! 😊