#include <stdio.h> // Incluye funciones de entrada/salida como printf y scanf

#include <stdlib.h> // Incluye funciones como malloc, free, rand, system

// -------------------- DEFINICIONES DEL PREPROCESADOR --------------------

// #define crea una constante simbólica o macro

#define PI 3.1416

#define CUADRADO(x) ((x)\*(x)) // Macro con parámetro (se reemplaza antes de compilar)

// -------------------- ESTRUCTURAS Y TIPOS --------------------

// struct permite agrupar diferentes tipos de datos en una sola entidad

struct Persona {

char nombre[50]; // Cadena de caracteres (arreglo de char)

int edad; // Edad en número entero

};

// enum define constantes enteras con nombres legibles

enum Dia { LUNES, MARTES, MIERCOLES };

// typedef permite crear alias para tipos

typedef struct Persona Persona;

// -------------------- FUNCIONES --------------------

// Declaración de una función con parámetros y valor de retorno

int sumar(int a, int b) {

return a + b; // return devuelve un resultado al llamador

}

// Función que no devuelve nada (void)

void mostrarMensaje() {

// Esta función solo realiza una acción sin devolver nada

}

// -------------------- FUNCIÓN PRINCIPAL --------------------

int main() {

// -------------------- DECLARACIÓN DE VARIABLES --------------------

int numero; // Variable entera

float decimal; // Variable con punto decimal

char letra; // Un solo carácter

char texto[100]; // Cadena de caracteres (arreglo)

const int CONSTANTE = 5; // Constante que no puede modificarse

// -------------------- OPERADORES --------------------

// Operadores se usan para calcular o comparar valores

// Aritméticos: +, -, \*, /, %

// Relacionales: ==, !=, <, >, <=, >=

// Lógicos: && (y), || (o), ! (no)

// De asignación: =, +=, -=, \*=, /=

// Incremento y decremento: ++, --

// -------------------- ESTRUCTURA if / else --------------------

// Permite ejecutar bloques condicionalmente

if (1 /\* condición booleana \*/) {

// Código que se ejecuta si la condición es verdadera

} else if (0 /\* otra condición \*/) {

// Se ejecuta si la anterior es falsa y esta es verdadera

} else {

// Se ejecuta si todas las anteriores condiciones son falsas

}

// -------------------- switch --------------------

// Permite ejecutar distintos bloques según el valor de una variable

enum Dia hoy = MARTES;

switch (hoy) {

case LUNES:

// Código si hoy es lunes

break; // Finaliza el caso

case MARTES:

// Código si hoy es martes

break;

default:

// Código si no coincide ningún caso

break;

}

// -------------------- BUCLE for --------------------

// Estructura de repetición con 3 partes

// for (inicio; condición; incremento) { cuerpo }

for (int i = 0; i < 10; i++) {

// i = 0; → se ejecuta al comenzar el bucle (una vez)

// i < 10; → condición para seguir repitiendo

// i++ → se ejecuta al final de cada iteración

}

// -------------------- BUCLE while --------------------

// Ejecuta el cuerpo mientras se cumpla la condición

int j = 0;

while (j < 5) {

// Código a ejecutar

j++; // Se debe actualizar manualmente

}

// -------------------- BUCLE do...while --------------------

// Igual que while, pero garantiza al menos una ejecución

int k = 0;

do {

// Código a ejecutar

k++;

} while (k < 5);

// -------------------- break y continue --------------------

// break: termina un bucle o switch

// continue: salta a la siguiente iteración

for (int n = 0; n < 10; n++) {

if (n == 5) continue; // Salta la iteración actual

if (n == 8) break; // Sale del bucle completamente

}

// -------------------- goto --------------------

// Salta a una etiqueta (se desaconseja por mala legibilidad)

goto salto; // Salta a la etiqueta llamada "salto"

// Código que se salta

salto:

// Punto al que se salta con goto

// -------------------- USO DE FUNCIONES --------------------

int resultado = sumar(3, 4); // Llama a la función sumar y guarda el resultado

mostrarMensaje(); // Llama a una función void

// -------------------- ESTRUCTURAS PERSONALIZADAS --------------------

Persona persona1 = {"Ana", 30}; // Uso de struct

// -------------------- MACROS --------------------

int lado = 5;

int area = CUADRADO(lado); // Reemplazado por ((lado)\*(lado)) al compilar

// -------------------- ENTRADA Y SALIDA --------------------

// printf: imprime texto

// scanf: lee valores desde el teclado

// printf("Introduce un número: ");

// scanf("%d", &numero); // Guarda lo que el usuario escribe en la variable

// -------------------- VALOR DE RETORNO DE main --------------------

return 0; // Código 0 indica que el programa terminó correctamente

}