C Programming

The C language can be considered a high level language and it facilitates a structured and disciplined approach to computer-program design.
C is a general-purpose, easy to learn and widely used in various applications.

by Hamdi Emad

Data Types and Variables

Fundamental Data Types

char : 1 byte
int, float : 4 bytes
double : 8 bytes

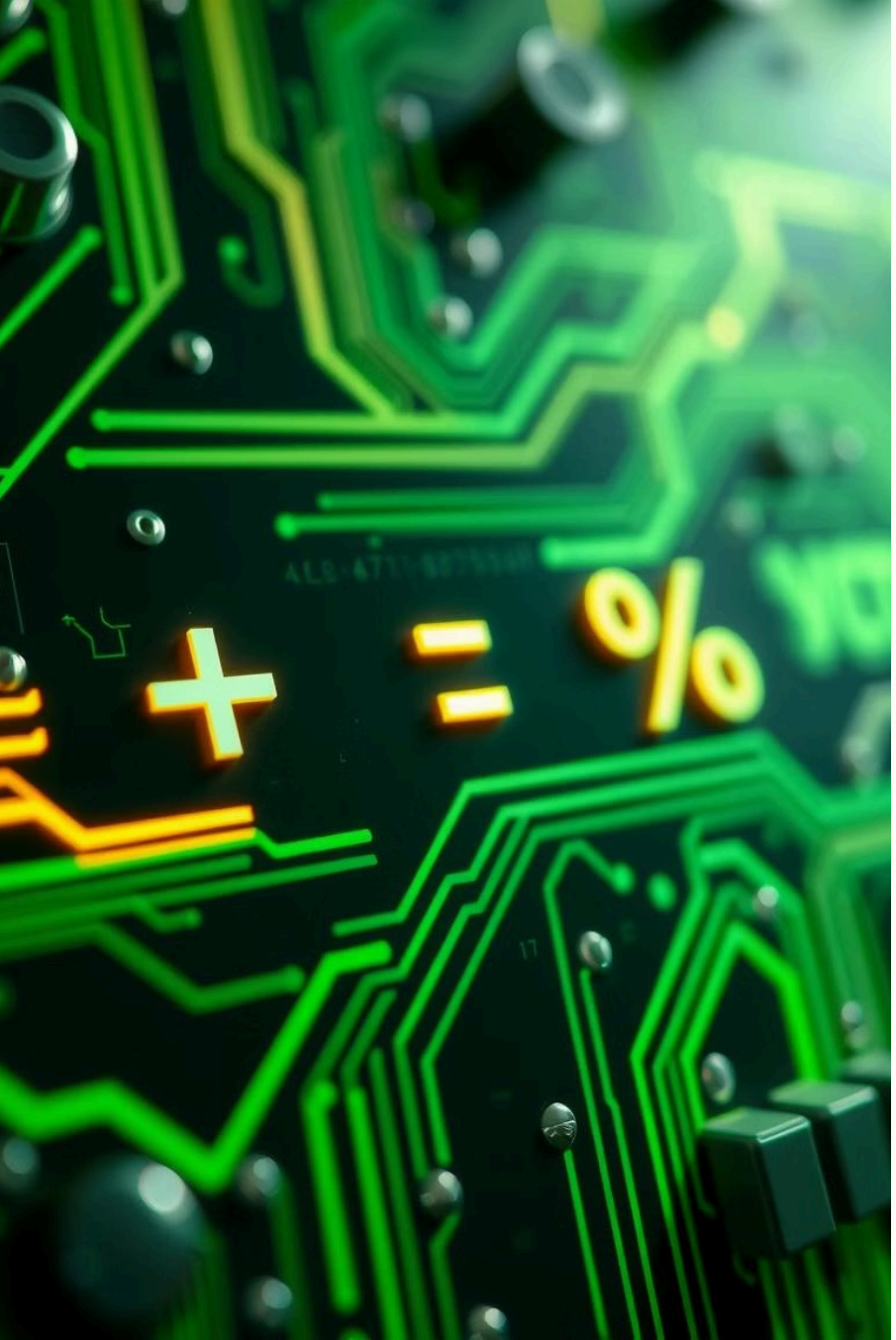
Variable Essentials

Declaration, initialization, and scope rules of each variable.

Type Qualifiers

const, volatile, static and extern enhance control.

Understanding these elements is crucial for effective C programming.



Operators and Expressions

Arithmetic

Perform calculations with different operators (+, -, *, /, %).

Relational

Compare values in conditions using (==, !=, >, <, >=, <=).

Logical

Combine conditions with (&&, ||, !).

Mastering operators is key to writing robust C code.

Control Flow Statements

Conditional

if, else if and else for decision-making.

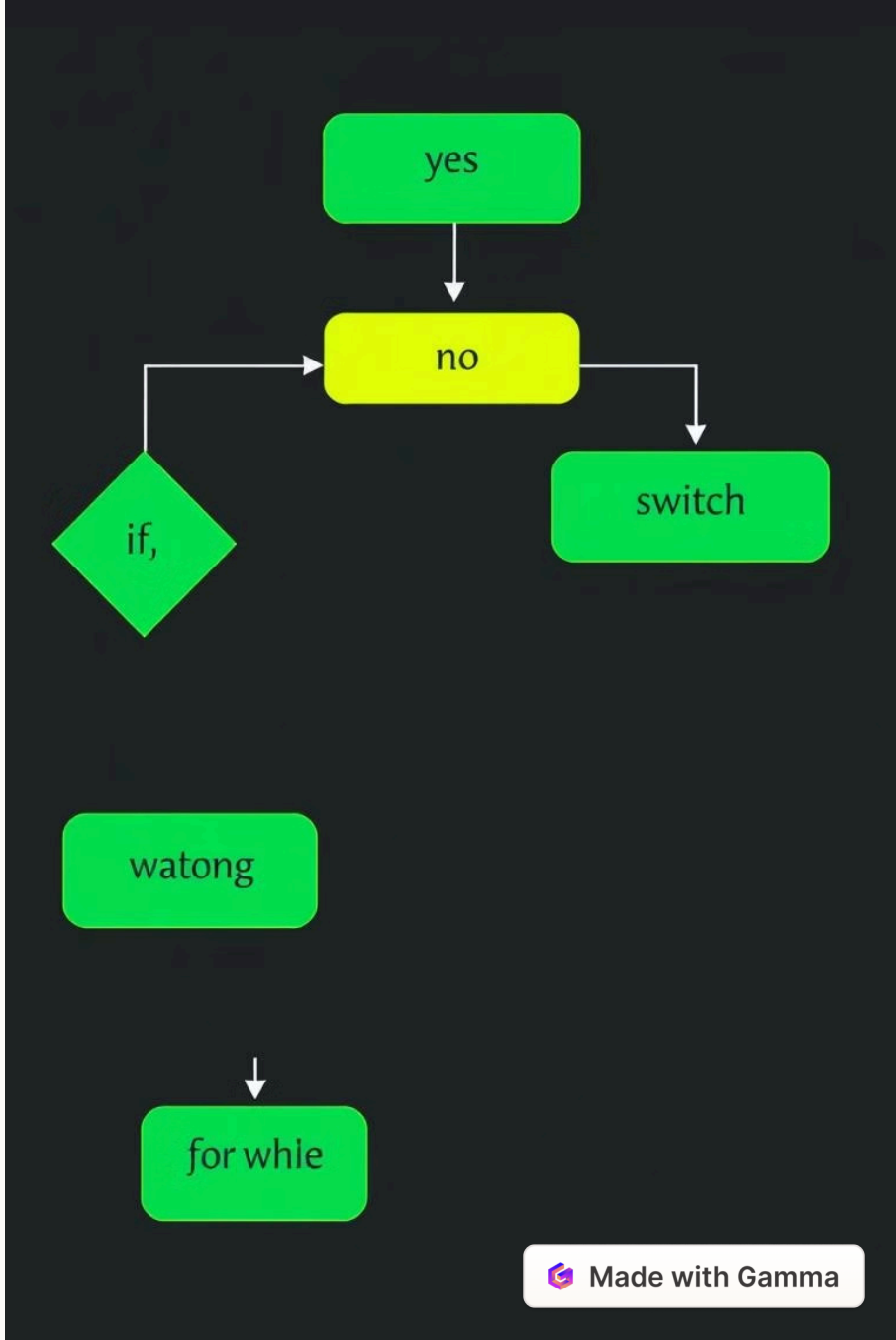
Switch

switch, case and default for multiple choices.

Looping

for, while and do-while for repetition.

Control flow is the backbone of program logic.



Functions

Declaration & Definition

Essential for modular programming.

Parameters & Return Types

Define function inputs and outputs.

Recursion

Solve problems by self-reference.

Functions promote code reusability and organization in C.

Arrays and Strings

Array Basics

Ordered collections of similar data.

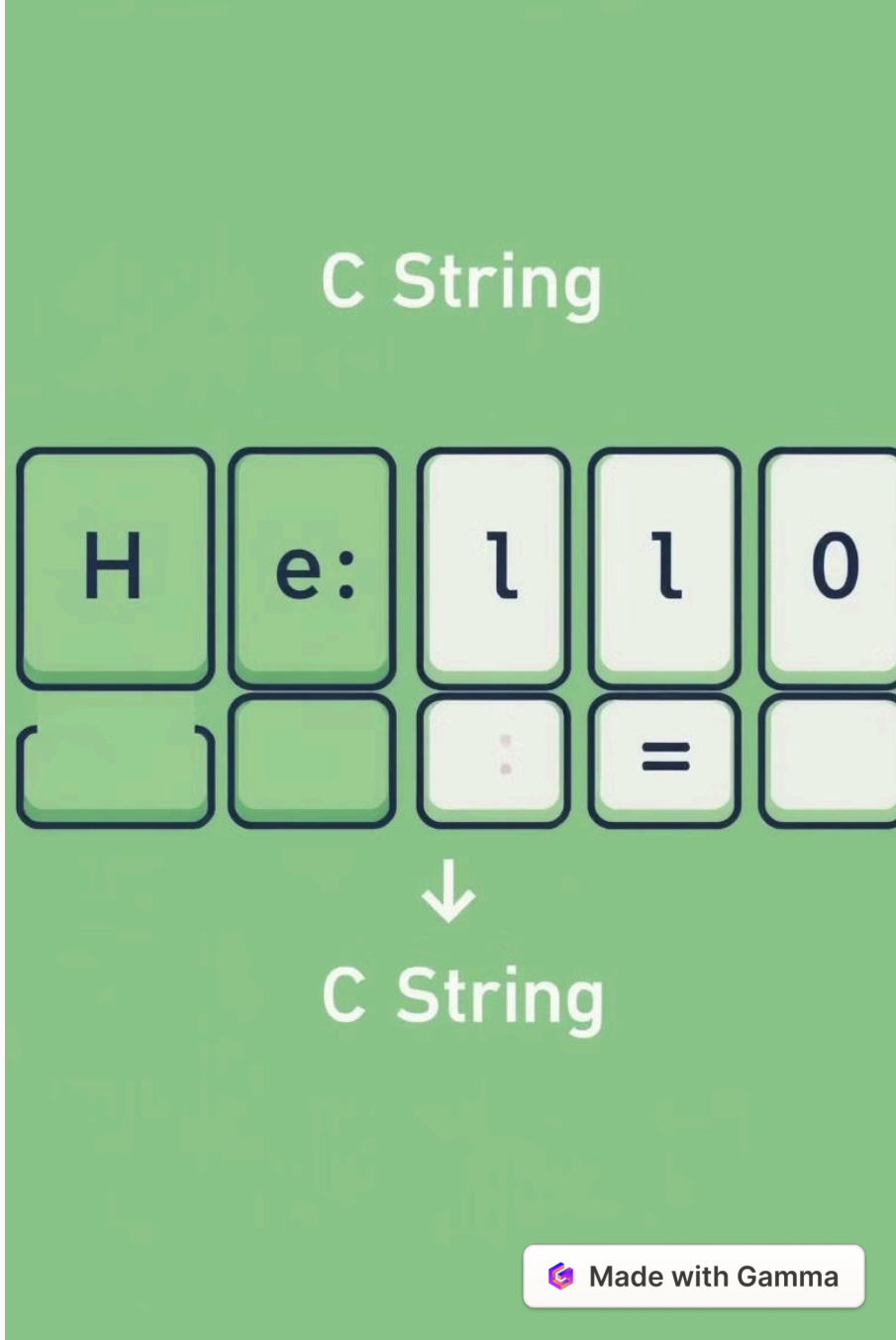
String Manipulation

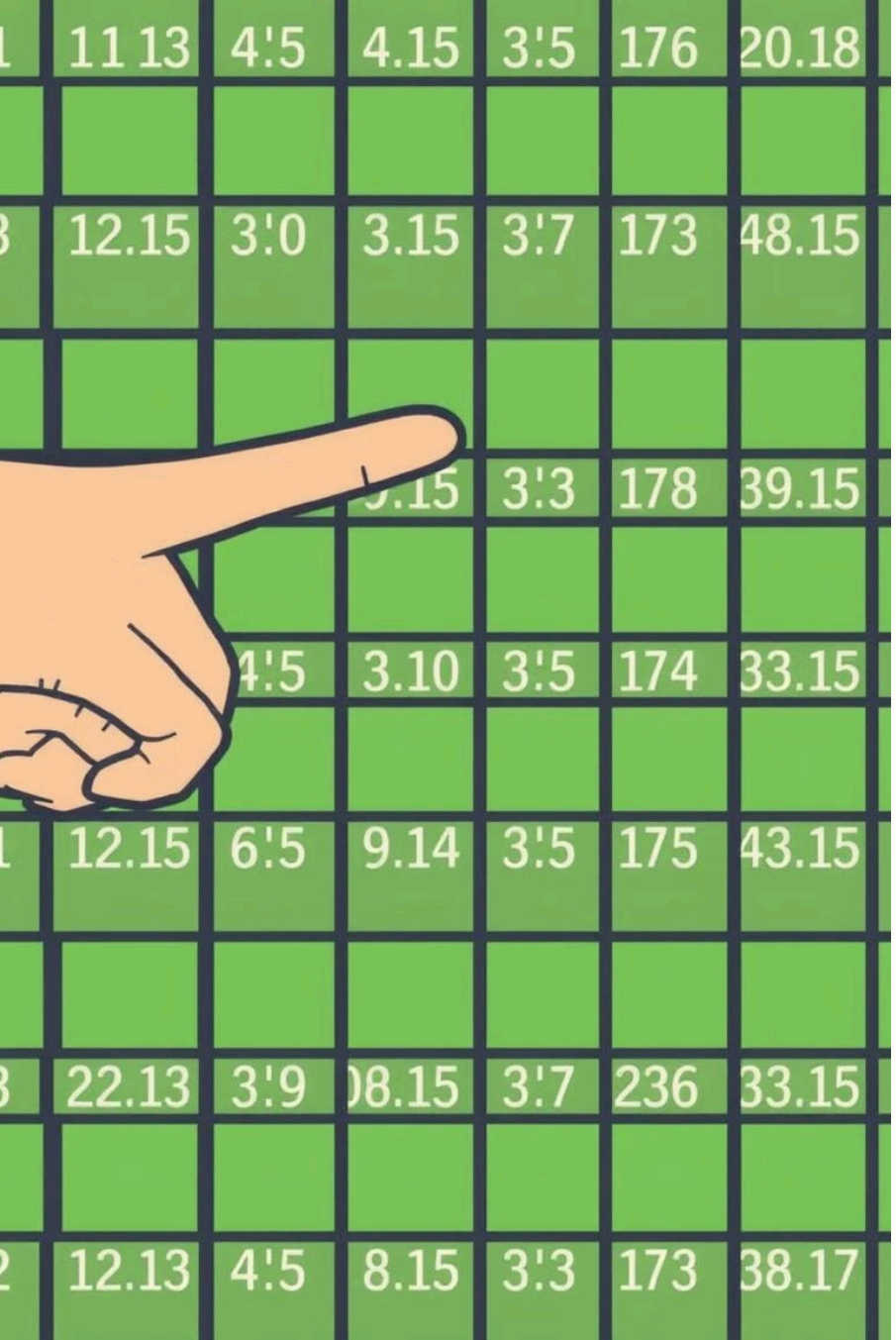
Use `` for common operations.

Null Termination

Strings end with a null character `\0`.

Arrays and strings are fundamental for data handling in C.



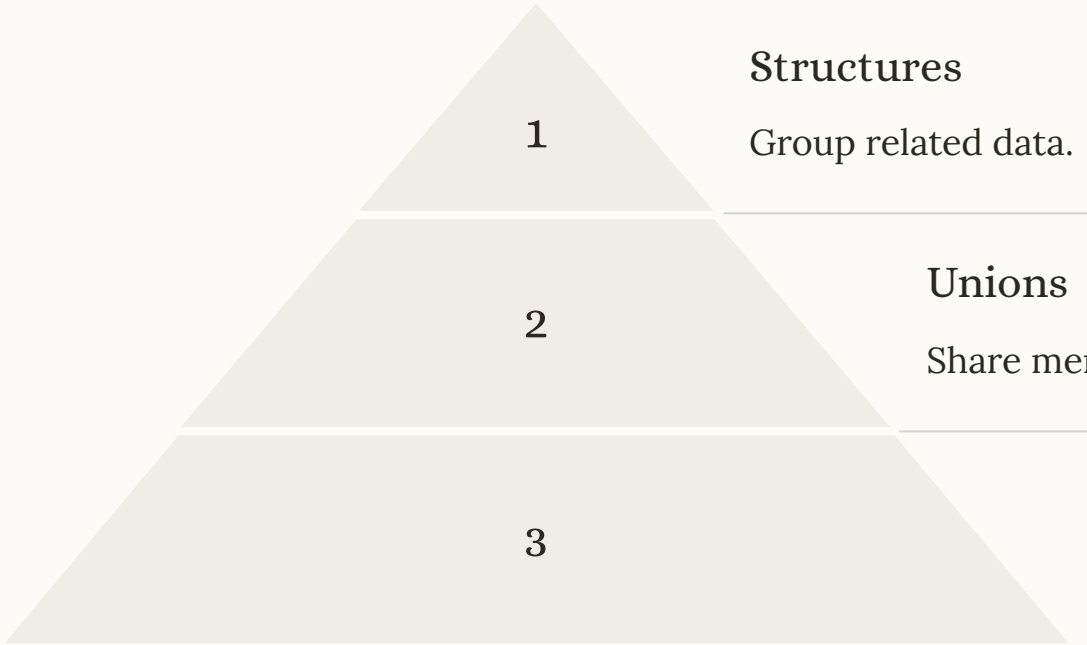


Pointers

- Declaration and dereferencing.
- Arithmetic operations.
- Dynamic memory allocation.

Pointers enable powerful memory manipulation in C.

Structures and Unions



Structures

Group related data.

Unions

Share memory locations.

File I/O

Reading and Writing files

Structures and unions allow creating custom data types in C.