Memory Diagrams & Representation

- Basic Diagram: Show variable name, value, type, and address.
- Endianness:
    - Little Endian: least significant byte at lowest address.
    - Big Endian: most significant byte at lowest address.
- Pointers: store addresses; use & (address-of) and * (dereference).
- 1D Arrays: arr[i] ≡ *(arr + i).

2D Arrays & Memory Allocation

- Stack 2D Array: int m[2][4]; → contiguous in row-major order.
- Heap 2D Array (Array of Arrays):
    - Allocate pointer to 2D, then allocate each 1D array inside. Free *all* including 2d pointer itself

Strings & <string.h>

- C-string: null-terminated (\0).
- Common functions: strlen, strcpy, strncpy, strcat, strcmp.
- Assigning a string literal to a stack-allocated array requires strcpy; direct assignment is invalid.

Structures

- Definition: can be local (struct name {}) or global (typedef struct {} name)
- Access:
    - Dot operator: myDate.month = 2;
    - Pointers: datePtr->year = 2025;
- Nesting & Arrays of Structs: struct members can be other structs or arrays.
- Passing Structs: by value (copy) or by pointer (efficient, can modify original).

I/O Overview

- Standard I/O (<stdio.h>):
    - Reading: scanf, getchar, fgets.
    - Writing: printf, putchar, puts.
    - Format Specifiers: %c, %s, %p, %x, %i, %f, %d.
- File I/O:
    - fopen, fclose, fprintf, fscanf, fgets, fputs.
    - Redirection in Linux: < input, > output, >> append.

C's Abstract Memory Model & Virtual Addressing

- Segments:
    1. CODE: .text, .rodata (read-only).
    2. DATA: .data (initialized) & .bss (uninitialized).
    3. HEAP: dynamic allocations (malloc, free).
    4. STACK: local variables, function calls.
- Global vs. Static Local: Both live in DATA segment. Global visible anywhere; static local persists across calls.
- 32-bit Process: typically 4GB virtual address space; user code vs. kernel space.

POSIX brk/sbrk & <unistd.h>

- brk(void *addr): sets top of heap to addr.
- sbrk(intptr_t incr): moves heap by incr bytes; returns old break or (void *)-1 on error.
- Avoid mixing these calls with malloc/free.

Heap Allocation Internals

- Functions: malloc, calloc, realloc, free.
    - Just exit with error message in cs354
- Block Structure: header (size + alloc bit), payload, (padding).
- Fragmentation:
    - External: gaps between free blocks.
    - Internal: unused space within allocated blocks.
- Allocator Approaches:
    - Implicit Free List: each block has size/alloc header, linear scan.
    - Explicit Free List: linked list of free blocks only.
- Placement Policies:
    - First Fit (FF): find first free block that fits.
    - Next Fit (NF): like FF but resumes from last position.
    - Best Fit (BF): smallest free block that fits (more thorough search)