Arrays and strings

What is an array?

- a collection of similar elements
  - ↳ They have the same type
- one dimension = vector
- two -D: matrix
  - ↳ also known as an array of vectors
- tensor: more dimensions
- arrays are sorted
- a[i] comes before [i -1] in memory

type name [count] { list }

type [count]

↳ unitiaized is [illegible] empty slots

m[x][y] → matrix

↳ stores them in row major order

↳ one row in memory followed by the next

memory

- 1-D
- rows are laid out seqentially

→ go baac and run the matrix mult function!

- arrays are not passed by vaies!
  - ↳ use call by refence

-we don-t want to copy large arrays into the stack

-arrays and pointer are similar

↳ a pointer to element zero of the array

& : memory location of a variable

size of 'function' = # of bytes used by a variable

a[0] = address a

a[i] = a + i · sizeof(a[0])

↳ pointers do the mult by sizeof

remember: a matrix is an array of pointers

String

-array of characters that end in '\0'

```
↳ char s[] = "blab"
  char *s = "hello"   → most common
  char s[] = {'L', ..}
```

in a nutshell, a string is an array of char that are terminated by a null char.

When you do j = s, they point to the same thing in the memory but it's not a copy strcmp= compares the strings together

strlen():- length of str

↳ num of non- null char

strcpy()°. make a copy & add a null at the end

strncmp(): make sure to add a check so you don't overflow