

macros.

\* Define Name of Macro Value  
 لى يكو استخراجه لى فى main

\* Define  $\text{add}(a, b) = (a + b)$

$\text{int } a = 5, b = 6;$

and say

$x = \text{add}(a, b)$

$x = a + b$

copied  
the value

we have also

we used it to make \* if del \* if del  
 sure "something" is we can write a code here  
 defined or not \* end if \* end if

macros & functions.

macros  $\rightarrow$  works with any type

$\hookrightarrow$  No overhead from function calls

Functions  $\rightarrow$  Type Safety & Proper error  
 $\hookrightarrow$  only evaluated once

creating



malloc, calloc, realloc.

Heap

malloc

P[5]

int \* P = (int\*) malloc(5 \* sizeof(int))

Calloc

int \* P = (int\*) calloc(5, sizeof(int))

realloc → to edit the size of malloc

int \* P = (int\*) realloc(P, 6 \* sizeof(int))

P[6]

and all of these we need to end with

free()

why?

Because of memory leaks

The system won't automatically reclaim memory for you

we can say \*A = int\* realloc(A, 0) = free(A)



## Headers

```
* ifndef My_Header_H
```

```
* def My_Header_H
```

```
// Declaration and definitions go here
```

```
* endif
```

Header guards prevent by ensuring the compiler processes the contents of a header file only once.