بسم الله الرحمن الرحيم

Marcos in C

التاريخ: / /

الموضوع

⇒ It is a symbolic name or const that represent value, expression or code snippet, It belongs to preprocessing phase

* Types of marcos :

① Object-like marco

ex. `#define PI 3.14`

```
int main(void){
   printf("%.2F", 1*1*PI);
}
```

② Chain marco (For more complex operations)

ex. `#define INSTGRAM FOLLOWERS`
`#define FOLLOWERS 138`

③ Function-Like marco :

ex. `#define add(x, y) (x+y)`

```
int main(void){
   printf("%d\n", add(2, 3));
}
```

SENA

بسم الله الرحمن الرحيم

التاريخ: / / Preprocessor directives الموضوع:

☆ Implemented before compile time, during preprocessing phase

1. #ifdef : used to check if a marco is defined or not

ex.
```
#define DEBUG
#ifdef DEBUG
   printf("debug mode is on");
#endif
```

2. #ifndef : used to check if a marco is not defined

```
#undef DEBUG
#ifndef DEBUG
   printf("debug mode is off");
#endif
```

3. #endif : used as a closing for #ifdef & #ifndef

4. #error : Trigger a compilation error

ex.
```
#ifndef CONFIG_SET
   #error "CONFIG_SET is missing"
#endif
```

5. #pragma : used for pack structs

ex. #pragma pack(1)

6. #undef : to undefine a marco that had been defined

7. #include : to include files and libraries

Dynamic memory Allocation

☆ In C , If you declared a variable , it will be stored in stack memory , So u can't change it size or delete it at run time , so dynamic memory allocation is used then

* malloc()
* calloc()
* realloc()
* free()

1. malloc() : allocates a block of contiguous memory at runtime

Note : Uninitialized

ex. `int *ptr = malloc ( sizeof (int) * 4);`

↳ array of int of size 4

⇒ `*(ptr + 1) = ptr[1]`

⇒ ptr = null if allocation fails

2. calloc() : similar to malloc() but it initializes the allocated memory to zero

ex. int *ptr = calloc(n , sizeof(data_type));
n: no. of blocks

int *ptr = malloc(sizeof(int)*5)

GV	GV	GV	GV	GV

↳ Garbage value

int *ptr = calloc(5 , sizeof(int))

0	0	0	0	0

3. free() : releases dynamically allocated memory by malloc , calloc

free(ptr); → ptr now called dangling pointer.

ptr = NULL // recommended

4. realloc() : used to resize a previously allocated memory block

realloc(ptr , new_size);

ex. int *ptr = malloc(sizeof(int) * 5);
ptr = realloc(ptr , 10 * sizeof(int));

Header guard in C :

Used in a header file (filename.h) to prevents multiple inclusion of the same header file

Syntax :

```
#ifndef FILENAME_H
#define FILENAME_H

// some code (marcos , enums, structs, function prototypes)

#endif // FILENAME_H
```

Why ??

main.C :

```
#include "filename.h"
```

filename.h

```
// headerguard
```

filename.c

```
#include "filename.h"
```

There is 2 #include "filename.h" so we need to prevent multiple inclusion by header guard.