# **Study about Macro in C**:

# Denifination:

A Macro is a fragment of code which has been given a name. When ever the name is called, it’s replaced by the contents of the Macro.

There are two type of Marco: object-like Macro and function-like Macro.

# Two kind of Macro:

## 2.1: Object-like Macro:

An object-like Macro is an simple identifer which will be replace by a code fragment. It looks like an data object in code that uses it.

* Creat an object-like Macro with the “#define” directive. “#define” if followed by the name of the Macro and then the token that is representative by the Macro. Ex: #define MAX\_VALUE 225
* In convention, Macro names are written in uppercase. Programe are easier to read when it is possible to tell at a glance which names are Macro.
* Macro is end at the end of the “#define” line. You can use character \ (bakcslash-newline) to continue the definition onto multiple lines.

## 2.2: Function-like Macro:

You can also define a Macro whose uses looks like a function call.

* To define a function-like Macro, use “#define” followed by the name of macro then is a pair of parentheses. Arguments are placed between that pair of parentheses.

Ex: #define PRINT\_X(x) \

printf(“Value of “ #x “is: %d“,x);

* Function-like Macro is only be expanded when it’s call with a pair of parentheses after it. If it’s just called by the name without (), it’s left alone.
* If you put spaces between Macro name and () it’s doesn’t define a funtion-like macro but a object-like macro whose expansion happens to begin with apair of parentheses.

# Macro argument.

* Function-like marco can take argument, just like true functions.
* To invoke a macro that take arguments, you write the name of the macro followed by the a list of actual argument is parenthese, separated by commas.
* The number of arguments you give must match with the number of parameters in the macro definition.
* When the macro is expaned each use of a paramater in its body is replaced by the tokens of correcsponding argument.

EX: #define MAX(x,y) (((x)>(Y))?(x):(y))

* You can leave macro argument empty.
* You cannot leave out the argument entirely, the number of comma between parentheses must be enough.

# Stringizing

* You can convert a macro argument to a string constant by using leading “#” preprocessing operator.
* When a macro parameter is used with leading “#”, the processor replace it with the literal text of the actual argument, converted to a string argument.

# Macro color

There is a kind of color call: “Macro color” which is color pencil, I bought an box of it conbining 36 color which cost 140k VND. Then I passed it to my close friend – MA with a price of 80k. She painted many picture using its, follow her instagram to watch them (https://www.instagram.com/mina.za.ra/).