**Procedure:**

A procedure is a named block of code (set of instructions) that is callable. Instead of rewriting the same piece of code multiple times (memory inefficient) we can simply call the procedure.

**Macros:**

A macro is also a named block of code that is called but it is expanded upon the call, that is, the code inside the macro is copied to the procedure inside which it was called, therefore, unlike a procedure, it is memory inefficient but since there is no stack involved it is time efficient.

UNLIKE PROCEDURES:

* Macros can accept direct parameters.
* Macros do not execute a return statement.
* Macro must be defined before it can be used.
* Time efficient
* Memory inefficient

**NOTES:**

Cannot pass parameter to the procedure directly but we can pass the parameter indirectly through:

1. Storing data in a variable
2. Storing data in a register
3. Through stack

When the procedure is called, the next immediate line after the procedure call is pushed to the stack so that the execution returns to that line after the procedure returns.

The RET statement inside the procedure POPS the last line from the stack that was pushed while calling the procedure and returns its control back to the main procedure.

Differences between macro and procedure:

1. macro is time efficient.
2. procedure is memory efficient.
3. macro does not return either indirectly or directly.
4. macro accepts parameters directly.
5. When the macro is used, the system copies the entire defined code. Therefore, macro is memory inefficient.

When the macro is called it is “Expanded”, that is, the copy of the statements is expended.

Also provides a convenient way to display a string.

**.MODEL SMALL**

**.STACK 100H**

**DISPLAY MACRO LINE**

MOV AH, 9

LEA DX, LINE

INT 21H

**ENDM**

**.DATA**

STR DB "Hello$"

**.CODE**

**MAIN PROC**

MOV AX, @DATA

MOV DS, AX

DISPLAY STR

MOV AH, 4CH

INT 21H

**ENDP MAIN**

**END MAIN**