**EXPERIMENT 9**

**AIM:** To study and implement Macros and Dos Interrupt in Assembly Language Programming.

**THEORY:**

**Macros:**

A macro is a sequence of instructions, assigned by a name and could be used anywhere in the program. Macros are useful for the following purposes:

* To simplify and reduce the amount of repetitive coding
* To reduce errors caused by repetitive coding
* To make an assembly program more readable.

A macro consists of name, set of formal parameters and body of code. The use of macro name with set of actual parameters is replaced by some code generated by its body. This is called macro expansion.

Macros allow a programmer to define pseudo operations, typically operations that are generally desirable, are not implemented as part of the processor instruction, and can be implemented as a sequence of instructions. Each use of a macro generates new program instructions, the macro has the effect of automating writing of the program.

Macros can be defined used in many programming languages, like C, C++ etc. Example macro in C programming.Macros are commonly used in C to define small snippets of code. If the macro has parameters, they are substituted into the macro body during expansion; thus, a C macro can mimic a C function. The usual reason for doing this is to avoid the overhead of a function call in simple cases, where the code is lightweight enough that function call overhead has a significant impact on performance.

**DOS Interrupts:**

The interrupt types 20h-3Fh are serviced by DOS routines that provide high-level service to hardware as well as system resources such as files and directories. The most useful is INT 21H, which provides many functions for doing keyboard, video, and file operations.

|  |  |
| --- | --- |
| **Function Number** | **Description** |
| AH=01h | READ CHARACTER FROM STANDARD INPUT, WITH ECHO |
| AH=02h | WRITE CHARACTER TO STANDARD OUTPUT |
| AH=05h | WRITE CHARACTER TO PRINTER |
| AH=06h | DIRECT CONSOLE OUTPUT |
| AH=07h | DIRECT CHARACTER INPUT, WITHOUT ECHO |
| AH=08h | CHARACTER INPUT WITHOUT ECHO |
| AH=09h | WRITE STRING TO STANDARD OUTPUT |
| AH=0Ah | BUFFERED INPUT |
| AH=0Bh | GET STDIN STATUS |
| AH=0Ch | FLUSH BUFFER AND READ STANDARD INPUT |
| AH=0Dh | DISK RESET |
| AH=0Eh | SELECT DEFAULT DRIVE |
| AH=19h | GET CURRENT DEFAULT DRIVE |
| AH=25h | SET INTERRUPT VECTOR |
| AH=2Ah | GET SYSTEM DATE |
| AH=2Bh | SET SYSTEM DATE |
| AH=2Ch | GET SYSTEM TIME |
| AH=2Dh | SET SYSTEM TIME |
| AH=2Eh | SET VERIFY FLAG |
| AH=30h | GET DOS VERSION |
| AH=35h | GET INTERRUPT VECTOR |
| AH=36h | GET FREE DISK SPACE |
| AH=39h | "MKDIR" - CREATE SUBDIRECTORY |
| AH=3Ah | "RMDIR" - REMOVE SUBDIRECTORY |
| AH=3Bh | "CHDIR" - SET CURRENT DIRECTORY |
| AH=3Ch | CREATE OR TRUNCATE FILE |
| AH=3Dh | "OPEN" - OPEN EXISTING FILE |
| AH=3Eh | "CLOSE" - CLOSE FILE |
| AH=3Fh | "READ" - READ FROM FILE OR DEVICE |
| AH=40h | "WRITE" - WRITE TO FILE OR DEVICE |
| AH=41h | "UNLINK" - DELETE FILE |
| AH=42h | "LSEEK" - SET CURRENT FILE POSITION |
| AH=43h | GET FILE ATTRIBUTES |
| AH=47h | "CWD" - GET CURRENT DIRECTORY |
| AH=4Ch | "EXIT" - TERMINATE WITH RETURN CODE |
| AH=4Dh | GET RETURN CODE (ERRORLEVEL) |
| AH=54h | GET VERIFY FLAG |
| AH=56h | "RENAME" - RENAME FILE |
| AH=57h | GET/SET FILE'S DATE AND TIME, GET EXTENDED ATTRIBUTES FOR FILE |

**CODE:**

**1) Program to calculate factorial using Macros.**

NUM: DW 0x6

RESULT: DW 0

MACRO fact(no) -> MUL word no <-

start:

MOV AX, 0x0001

NOTZEROLOOP:

fact(NUM)

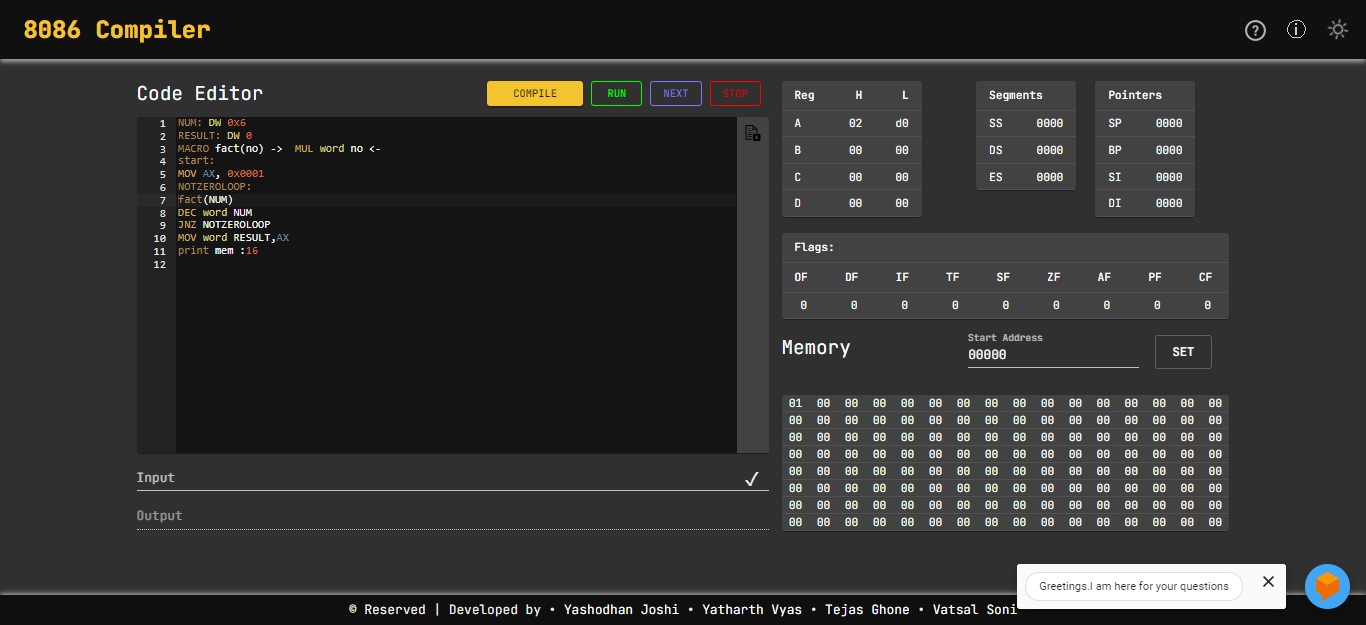
DEC word NUM

JNZ NOTZEROLOOP

MOV word RESULT,AX

print mem :16

**OUTPUT:**



**2) Program to calculate the sum of 2 number using DOS Interrupts.**

DATA SEGMENT

NUM1 DB ?

NUM2 DB ?

RESULT DB ?

MSG1 DB 10,13,"ENTER FIRST NUMBER TO ADD : $"

MSG2 DB 10,13,"ENTER SECOND NUMBER TO ADD : $"

MSG3 DB 10,13,"RESULT OF ADDITION IS : $"

ENDS

CODE SEGMENT

ASSUME DS:DATA, CS:CODE

START:

MOV AX,DATA

MOV DS,AX

LEA DX,MSG1

MOV AH,9

INT 21H

MOV AH,1

INT 21H

SUB AL,30H

MOV NUM1,AL

LEA DX,MSG2

MOV AH,9

INT 21H

MOV AH,1

INT 21H

SUB AL,30H

MOV NUM2,AL

ADD AL,NUM1

MOV RESULT,AL

MOV AH,0

AAA

ADD AH,30H ADD AL,30H

MOV BX,AX

LEA DX,MSG3

MOV AH,9

INT 21H

MOV AH,2

MOV DL,BH

INT 21H

MOV AH,2

MOV DL,BL

INT 21H

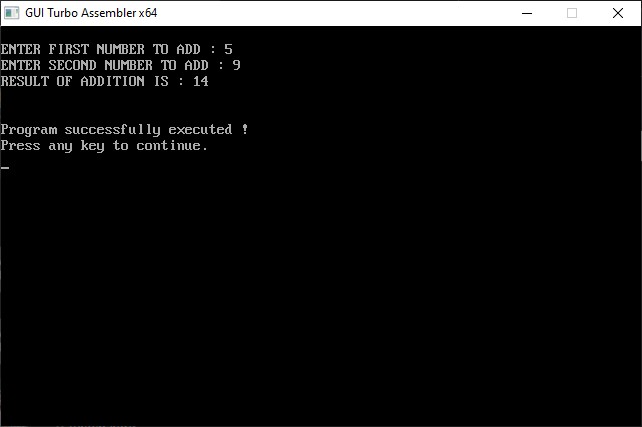
MOV AH,4CH

INT 21H

ENDS

END START

**Output:**



**CONCLUSION:**

In this experiment, I implemented Macros and Dos Interrupts in Assembly language Program.