**PRACTICAL 14**

**Aim:** Write an assembly program to check given number is even or odd. Also print appropriate message on console.

**Description of instructions used:**

**MSG:** Here MSG stands for message here in place of MSG we will also use STRING.

**DB** **(DEFINE BYTE):** The **DB** directive is used to declare a byte type variable, or a set aside one or more storage locations of type byte in memory.

**MARCO:** A **Macro** is a set of instructions grouped under a single unit. It is another method for implementing modular programming in the **8086** microprocessors (The first one was using Procedures)

**LEA (Load Effective Address):** LEA and MOV both are same but in that there are quite difference between both of them.

* LEA means Load Effective Address
* MOV means Load Value

In short, LEA loads a pointer to the item you're addressing whereas MOV loads the actual value at that address. The purpose of LEA is to allow one to perform a non-trivial address calculation and store the result.

**LEA AX, [BP+SI+5]; Compute address of value**

**MOV AX, [BP+SI+5]; Load value at that address**

**ASSUME:** The ASSUME directive tells the assembler to assume, that a certain register contains the base of some structure (in your case: segments). In your case, CS and DS point to the code segment and the data segmentrespectively, both the one and only of their respective kind.

**DISPLAY:** For display SRTING or in here MSG.

**INT:** INT is an assembly language instruction for x86 processors that generates a software interrupt. It takes the interrupt number formatted as a byte value.

**CMP:** The CMP instruction compares two operands. It is generally used in conditional execution. This instruction basically subtracts one operand from the other for comparing whether the operands are equal or not. It does not disturb the destination or source operands.

**JNE:** The JNE (or JNZ) instruction is a conditional jump that follows a test. It jumps to the specified location if the Zero Flag (ZF) is cleared (0). JNZ is commonly used to explicitly test for something not being equal to zero whereas JNE is commonly found after a CMP instruction.

**JMP:** the JMP instruction performs an unconditional jump. Such an instruction transfers the flow of execution by changing the instruction pointer register.

**Code:**

DATA SEGMENT

INPUTSTR DB 10,13,'ENTER NO $'

EVENSTR DB 10,13,'EVEN NUMBER $'

ODDSTR DB 10,13,'ODD NUMBER $'

DATA ENDS

DISPLAY MACRO MSG

MOV AH,9

LEA DX,MSG

INT 21H

ENDM

CODE SEGMENT

ASSUME CS: CODE, DS: DATA

START:

MOV AX, DATA

MOV DS, AX

DISPLAY INPUTSTR

MOV AH,1

INT 21H

MOV AH,0

CHECK:

MOV DL,2

DIV DL

CMP AH,0

JNE ODD ;JUMP IF ZERO FLAG IS CLEARED

EVEN:

DISPLAY EVENSTR

JMP DONE

ODD:

DISPLAY ODDSTR

DONE:

MOV AH,4CH

INT 21H

CODE ENDS

END START

**Output:**

