**Practical - 21**

**Aim:** Write an assembly program to check given number is prime or not. Display ‘1’ on output screen if number is prime, display ‘0’ otherwise.

**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:**

**.MODEL PRIME**

**.DATA**

**MSG DB "The Give No is a Prime No$"**

**NMSG DB "The Given No is not a Prime No$"**

**NUM DB 70H ;Enter the required no here**

**.CODE**

**START:**

**MOV AX,@DATA**

**MOV DS,AX**

**MOV AL,NUM**

**MOV BL,02H** ; The Dividing starts from 2, Hence BH is compare to 02H

**MOV DX,0000H**  ; To avoid Divide overflow error

**MOV AH,00H** ; To avoid Divide overflow error

;Loop to check for Prime No

**L1:**

**DIV BL**

**CMP AH,00H** ; Remainder is compared with 00H (AH)

**JNE NEXT**

**INC BH** ; BH is incremented if the Number is divisible by current value of BL

**NEXT:CMP BH,02H**; If BH > 02H, There is no need to proceed, It is not a Prime

**JE FALSE** ; The no is not a Prime No

**INC BL** ; Increment BL

**MOV AX,0000H** ; To avoid Divide overflow error

**MOV DX,0000H** ; To avoid Divide overflow error

**MOV AL,NUM** ; Move the Default no to AL

**CMP BL,NUM** ; Run the loop until BL matches Number. I.e, Run loop x no of times, where x is the Number given

**JNE L1**  ; Jump to check again with incremented value of BL

;To display The given no is a Prime No

**TRUE:**

**LEA DX,MSG**

**MOV AH,09H ;** Used to print a string

**INT 21H**

**JMP EXIT**

;To display The given no is not a Prime No

**FALSE:**

**LEA DX,NMSG**

**MOV AH,09H**; Used to print a string

**INT 21H**

**EXIT:**

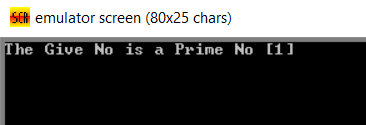
**MOV AH,4CH**

**INT 21H**

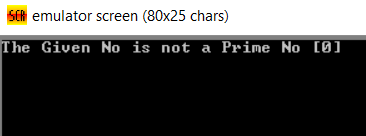
**END START**

**Output:**

**INPUT NO: 71H**



**INPUT NO: 70H**

\