**Practical - 3**

**Aim:** Write an assembly program for below given fragments of C program

1. void main ()

{

int l,m,n,o,p;

l = m+n-o+p;

}

1. C = (F – 32) \* 5 / 9

**Description of instructions used:**

**MOV:**

* The MOV instruction copies a word or byte of data from a specified source to a specified destination. The destination can be a register or a memory location. The source can be a register, a memory location or an immediate number. The source and destination cannot both be memory locations. They must both be of the same type (bytes or words). MOV instruction does not affect any flag.
* Syntax:

mov <reg>,<reg>

mov <reg>,<mem>

mov <mem>,<reg>

mov <reg>,<const>

mov <mem>,<const>

**ADD:**

* These instructions add a number from some source to a number in some destination and put the result in the specified destination.
* Syntax:

add <reg>,<reg>

add <reg>,<mem>

add <mem>,<reg>

add <reg>,<const>

add <mem>,<const>

**SUB:**

* These instructions subtract the number in some source from the number in some destination and put the result in the destination.
* Syntax:

sub <reg>,<reg>

sub <reg>,<mem>

sub <mem>,<reg>

sub <reg>,<const>

sub <mem>,<const>

**MUL:**

* This instruction multiplies an unsigned byte in some source with an unsigned byte in AL register or an unsigned word in some source with an unsigned word in AX register. The source can be a register or a memory location.
* Syntax:

mul <reg>,<reg>

mul <reg>,<mem>

mul <mem>,<reg>

mul <reg>,<const>

mul <mem>,<const>

**DIV:**

* This instruction is used to divide an unsigned word by a byte or to divide an unsigned double word (32 bits) by a word. When a word is divided by a byte, the word must be in the AX register. The divisor can be in a register or a memory location.
* Syntax:

div <reg>,<reg>

div <reg>,<mem>

div <mem>,<reg>

div <reg>,<const>

div <mem>,<const>

**Code:**

**1)**

.model prac3

.data

l DB 1 DUP<?>

m DB 1 DUP<?>

n DB 1 DUP<?>

o DB 1 DUP<?>

p DB 1 DUP<?>

.code

MOV AX,DATA

MOV DS,AX

MOV AL,m

MOV BL,n

ADD AL,BL

MOV BL,o

SUB AL,BL

MOV BL,p

ADD AL,BL

MOV l,AL

END

**2)**

.model prac3p2

.data

C DB 1 DUP<?>

F DB 1 DUP<?>

.code

MOV AX,DATA

MOV DS,AX

MOV AL,F

MOV BL,32H

SUB AL,BL

MOV BL,5H

MUL BL

MOV BL,9H

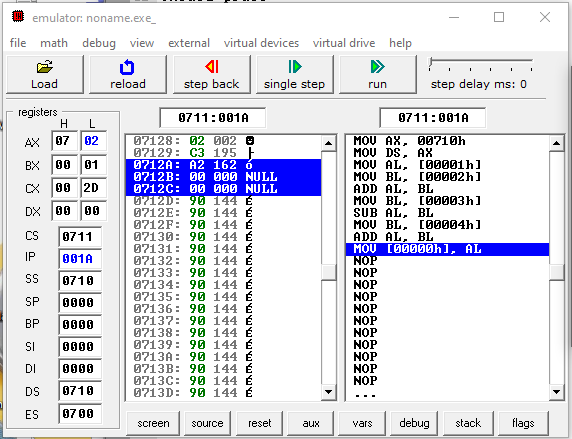
DIV BL

MOV C,AL

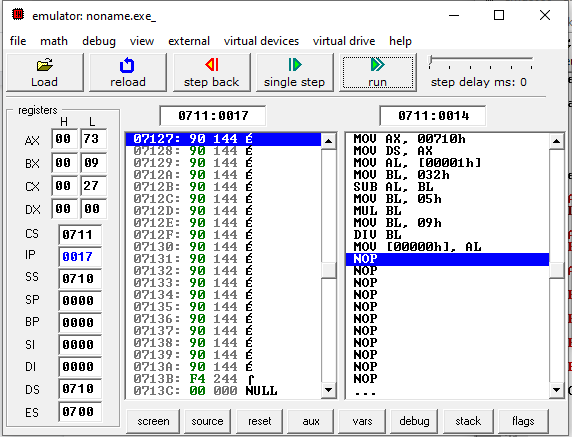
END

**Input/output:**

**1)**



**2)**

****