**Lab Practice Exercises**

;<Author> : Nitin

;Date - 21.03.2022

;Description - Division

jmp start ; Jump to the place where the label "start"

;code

start: nop ; <LABEL:> Instruction Operands

mvi a, 0 ; move the value 0 to A register

mvi b, 0 ; move the value 0 to B register

mvi c, 0 ; move the value 0 to C register

mvi a, 8 ; move the value 0 to A register

mvi b, 2 ; move the value 0 to B register

mvi c, 0 ; move the value 0 to C register

loop: inr c ; Increment the memory

sub b ; Decrement the counter

jnz loop ; Check again

hlt

;<Author> : Nitin

; Date - 28.03.2022

; Description - Smallest of Numbers

jmp start ; Jump to the place where the label "start"

;code

start: nop ; <LABEL:> Instruction Operands

lxi h, 0001H ; Load the memory address where the list starts

mov b,m ; move the content from the memory to B register

mvi d,3 ; Number of value to compare

loop: mov a,m ; Move the content to accumulator

cmp b ; compare the value in accumulator with B Register

jnc skip ; skip if greater

mov b,a ; if smaller move to B

skip: inx h ; Increment the memory

dcr d ; Decrement the counter

jnz loop ; Check again

hlt

;<Author> : Nitin

; Date - 4.04.2022

; Description – Factorial\_Demo

jmp start ; Jump to the place where the label "start"

; code

start: nop ; <LABEL:> Instruction Operands

lda 0006 ;load value in 0006 to accumulator move value in accumulator to D

mov D, A

loop: der D ; decrement the value in D by 1

jz skip ;jump to lable skip if the value in accumulator is 0

mul D ;multiply value in accumulator with value in D

jmp loop ; jump to lable loop

skip: sta 0008 ; store value in accumulator to 0008

hlt ; The value in 0008 is the final value (factorial of the given number).

;<Author> : Nitin

; Date - 11.4.22

; Description - Factorial

jmp start

;code

start: nop

lxi h, 0003H ; Load the memory address where the list starts

mov b,m ; move the content from the memory to B register

mov c,b ; move the content from the B to C register

dcr b ; decrement the B register

loop1: mov d,b ; move the content from the B to D register

mvi a,00 ; move the content from the B to C register

loop2: add c ; add the value in C to the accumulator

dcr d ; decrement the D register

jnz loop2 ; check again

mov c,a ; move the content from the A to C register

dcr b ; decrement the B register

jnz loop1 ; jump to loop1

hlt