

EX 1:

decrement counter (10 to 0) [if No memory location]

Simple program:

MVI C, 0AH

Back: DCR C

JNZ BACK ; if result $\neq 0 \Rightarrow C \neq 0$ jump

HLT

- Program that loads the value 10 into a memory location, then decrement it in a loop until it reaches 0.

MVI A, 0AH

; [A] $\leftarrow$ 10

STA 2020H

[2020] $\leftarrow$ [A]

LXI H, 2020H

H/L points to memory address 2020H

Loop: MOV A, M

; A $\leftarrow$ M[2020]

DCR A

; decrement A (A = A - 1)

STA 2020H

; Store the decremented value back to memory.

CPI 00H

; Compare A with 00H

JNZ loop

; if A $\neq 0$, jump back to loop.

HLT;

Stop execution.

19 Bytes : Total size of prog.

EX 2: Add 5 consecutive numbers starting from memory address 3000H. Store the final result at memory location 3050H.

LXI H, 3000H

; H/L points to the first number H = 30H L = 00H

MVI C, 05H

; C $\leftarrow$ 05H (counter)

MVI A, 00H

A $\leftarrow$ 00H

Loop: ADD M

; A $\leftarrow$ A + [HL]

INX H

HL $\leftarrow$ HL + 1 (pointed the next address)

DCR C

C $\leftarrow$ C - 1 (Mise à jour flags sauf carry)

JNZ loop

; if result $\neq 0$ (C $\neq 0$) return to loop

STA 3050H

store result in [3050]

HLT;

PC = 0011H

[3050]H = final value [A]

A = Somme des 5 bytes

C = 00H ; finish loop

HL = 3005H (après 5 adresses)

C = 00H ; suivant s'il y a un dépassement

Ex3: The Subroutine must use PUSH and POP to save registers (LXI SP, FFFFH)

2 Bytes MVI A, 25H 0000H

3 Bytes CALL COMPLEMENT 0002H

3 Bytes STA 01AH 0005H

1 Byte HLT 0008H

← l'adresse qui doit être sauvegardée SP.

1 Byte COMPLEMENT: CMA ; complemented A

1 Byte RET 0009H 000AH

11 Bytes: size total PC: 000BH

Push PSW → Save A and flags before call: if SP = FFFFH

POP PSW → Restore A and flags after call

FFFFB → address memory constant
FFFFCH → [65] flags
FFFDH → [2] = A
FFFEH
FFFFH

FFFFC
FFFD 05H
FFFE 00H
FFFFH

High

CALL COMP (SP = SP - 2) → FFFDH

RET (SP = SP + 2) → FFFDH

SP-1: FFFEH = PCH(00H)
SP-2: FFFDH = PCL(05H)
FFFDH 05H
FFFEH 00H

Low

Ex4 example: 010DH { FFFD → 10
FFFE → 20
FFFF

LXI H, 0205H

HL ← 0205H

MVIC, 0AH

; C: counter loop 10

MVI B, 00H

B ← 00H (0 nombre positif)

loop1: MOVA, M

[A] ← [M_{HL}]

ANI 80H (AND [A] 80H) Masque deux bits sans le bit de signe

JNZ loop2

if bit 7 (signe) = 1, jump l'incrément

INR B

if positif increment counter (B: nombre positif)
; Next Number

loop2: INX H

DCR C

Decrement counter C

JNZ loop1

if (result ≠ 0) C (≠ 0) continue Loop

MOV A, B

total number positif in A

STA 0220H

store result in 0220H

HLT

AND logic: example: A = 7FH → 0111 1111 and 1000 0000
A = 55H → 0101 0101 and 1000 0000
↑ positif
↑ négatif