**Enunciado:** Programa que realiza la suma o resta de dos datos DA1 (78H) y DA2 (79 H) dependiendo de si es cero (suma) o no (resta) la posición de memoria CHE (77H). El resultado se guarda en la localidad de memoria 7AH.

**Objetivo:** Estructura de una decisión.

**Código**:

CHE EQU 77H

DA1 EQU 78H

DA2 EQU 79H

RES EQU 7AH

ORG 0000H

SJMP INICIO

ORG 0040H

INICIO: MOV DA1,#09H

MOV DA2, #06H ; Carga los datos en memoria

MOV A,CHE ; Lee la posición de memoria 77H

JNZ SAL1 ; Si es igual a cero realiza la suma y si no,

; salta a efectuar resta

MOV A, DA1 ; Realiza suma y guarda resultado

ADD A, DA2

MOV RES, A

MOV A, CHE

JZ SAL2 ; Salta código de la resta

SAL1: MOV A,DA1

SUBB A,DA2

MOV RES,A ; Realiza resta y guarda resultado

SAL2: NOP

END

**Enunciado:** La posición de memoria MEM (77H) contiene un número entre 1 y 9; se cargaran a FFh tantas localidades de memoria a partir de la 61 H como se indique en MEM.

**Objetivo:** Estructura al menos una vez.

**Código:**

MEM EQU 77H

CON EQU 78H

ORG 0000H

SJMP INICIO

ORG 0040H

INICIO: MOV CON,MEM ; Guarda el valor de MEM

; en nua variable llamada CON

SAL: MOV A,CON ;

ADD A,#60H ; Prepara indexado

MOV R0,A

MOV @R0,#0FFH ; Carga FF a donde indica R0

DJNZ CON, SAL

NOP

END

**Enunciado:** Repetir el programa anterior pero MEM puede contener un cero,

**Objetivo:** Estructura iterativa puede que ni una vez (revisa condición primero).

**Código:**

MEM EQU 77H

CON EQU 78H

ORG 0000H

SJMP INICIO

ORG 0040H

INICIO: MOV CON,MEM ; Guarda el valor de MEM

; en nua variable llamada CON

SAL1: MOV A,CON ;

JZ SAL2 ; Salta al final si no se

; va a escribir ningún dato

ADD A,#60H ; Prepara indexado

MOV R0,A

MOV @R0,#0FFH ; Carga FF a donde indica R0

DEC CON

SJMP SAL1

SAL2: NOP

END

**Enunciado**: Si el bit 2 de la posición de memoria CHE (28H) es igual a cero, entonces la posición de memoria RES (70H) se señalará poniendo todos su bits a cero; en caso contrario, a unos.

**Objetivo:**  Comprobar bits. Método general.

**Código:**

CHE EQU 28H

RES EQU 70H

ORG 0000H

SJMP INICIO

ORG 0040H

INICIO: MOV A,CHE

ANL A, #04H ; Aisla el segundo bit

JZ SAL ; Si es igual a cero, salta

MOV RES, #0FFH ; Si no carga RES con unos

SAL: JNZ FIN

MOV RES,#00H ; Carga RES con ceros

FIN: NOP

END

**Enunciado:** Realizar el programa anterior auxiliándonos con el carry.

**Objetivo:** Comprobar un bit empleando el carry.

**Código:**

CHE EQU 28H

RES EQU 70H

ORG 0000H

SJMP INICIO

ORG 0040H

INICIO: SETB C ; Poner carry flag en 1

ANL C,CHE.2 ; Comprobar valor del bit deseado

JNC SAL1 ; Si es cero, salta

MOV RES,#0FFH ; Carga RES con unos

SAL1: JC FIN ;

MOV RES,#00H ; Carga REs con ceros

FIN: NOP

END