

**08) Rotación a izquierda de bits – Por registro y valor**

ORG 3000H

; Precondición: el byte está en el registro AL, y la cantidad de rotaciones en AH

```

ROTARIZ:    CMP AH, 0
               JZ FIN
               ADD AL, AL           ; Multiplicación por 2
               JNC NEXT
               INC AL               ; Incremento por acarreo
NEXT:      DEC AH                 ; Rotaciones restantes
               JMP ROTARIZ
FIN:       RET

```

**12) Rotación a derecha de bits – Por registro y valor**

ORG 3100H

; Precondición: el byte está en el registro AL, y la cantidad de rotaciones en CL

```

ROTARDER:  MOV AH, 8
               SUB AH, CL           ; Equivalencia Izquierda - Derecha
               JNS OK
FIX:      ADD AH, 8             ; Caso rotaciones a derecha mayor a 8
               JS FIX
OK:       CALL ROTARIZ
               RET

```

**09) Conteo de caracteres terminado en 0 – Por registro y referencia**

ORG 3000H

; Precondición: la dirección de inicio de la cadena está en el registro BX

```

CONCAR:    PUSH BX
               MOV AL, 0
LOOP:     CMP BYTE PTR [BX], 0
               JZ FIN
               INC AL             ; Incremento del contador
               INC BX             ; Siguiente carácter
               JMP LOOP
FIN:      POP BX
               RET               ; Devuelvo conteo en AL

```

**11) Conteo y reemplazo de un carácter dado – Por registros**

ORG 3000H

; Precondición: la dirección de la cadena está en BX, y el carácter en CL

```

CONCAR:    PUSH BX
               MOV AL, 0
LOOP:     MOV CH, BYTE PTR [BX]
               CMP CH, 0
               JZ FIN
               CMP CH, CL         ; Verifico si coincide carácter
               JNZ NEXT
               MOV BYTE PTR [BX], 88
               INC AL             ; Incremento del contador
NEXT:     INC BX             ; Siguiente carácter
               JMP LOOP
FIN:      POP BX
               RET             ; Devuelvo conteo en AL

```

**10) Intercambio de palabras entre sí – Por pila y referencia**

ORG 3000H

; Precondición: se apilaron las direcciones de las palabras W1 y W2

<b>SWAP:</b>	<b>PUSH BX</b> <b>MOV BX, SP</b> ; Puntero a pila <b>PUSH CX</b> <b>PUSH AX</b> <b>PUSH DX</b> <b>ADD BX, 4</b> ; Puntero a Stack OFFSET W2 <b>MOV AX, BX</b> ; Puntero copiado en AX <b>MOV BX, [BX]</b> ; Indirección: BX es OFFSET W2 <b>MOV DX, [BX]</b> ; Indirección: DX es W2 <b>MOV BX, AX</b> ; Restauro puntero a pila <b>ADD BX, 2</b> ; Puntero a Stack OFFSET W1 <b>MOV BX, [BX]</b> ; Indirección: BX es OFFSET W1 <b>MOV CX, [BX]</b> ; Indirección: CX es W1 <b>MOV [BX], DX</b> ; Reemplazo W1 por W2 <b>MOV BX, AX</b> ; Restauro puntero a pila <b>MOV BX, [BX]</b> ; Indirección: BX es OFFSET W2 <b>MOV [BX], CX</b> ; Reemplazo W2 por W1 <b>POP DX</b> <b>POP AX</b> <b>POP CX</b> <b>POP BX</b> <b>RET</b>
--------------	---