Práctica 2: E/S por consulta de estado (PIO y HAND-SHAKE)

Parte 1: Entrada/Salida con Luces y Llaves mediante el PIO

Ejercicio 1

a)

```
PB    EQU 31h          ; constantes del ensamblador
CB    EQU 33h          ; 1° línea faltante: registro CB para configurar
                       ; el puerto B del PIO

      ORG 2000H
      mov al, 0        ; 2° línea faltante
      out CB, al       ; configurar todos los conectores del puerto
                       ; B del PIO (luces) como salidas
      mov al, 0Fh      ; 0Fh = 00001111b
      out PB, al       ; encender las luces 3 a 0 y apagar las restantes
                       ; (7 a 4)
      int 0
      end
```

b)

```
PA    EQU 30h          ; 1° línea faltante: registro de datos PA del
                       ; puerto A del PIO
CA    EQU 32h          ; registro de configuración del puerto A del PIO

      ORG 1000h
msj   db "Apagadas"

      ORG 2000H
      mov al, 0FFh     ; 0FFh = 11111111b
      out CA, al       ; 2° línea faltante: configurar todos los
                       ; conectores del puerto A (llaves) como entradas
      in al, PA        ; leer y obtener el estado actual de las llaves
      cmp al, 0
      jnz fin          ; si al menos una de las llaves se encuentra
                       ; prendida, finalizar inmediatamente el programa
      mov al, 8
      mov bx , offset msj
      int 7            ; en caso contrario, imprimir en pantalla
                       ; el mensaje "Apagadas" y luego finalizar
fin:  int 0
      end
```

c)

```
PA    EQU 30h          ; 1° línea faltante
PB    EQU 31h
CA    EQU 32h
CB    EQU 33h          ; 2° línea faltante
```

```
ORG 2000h
mov al, 0FFh     ; 3° línea faltante: 0FFh = 11111111b
out CA, al       ; 4° línea faltante: configurar todos los
                 ; conectores del puerto A (llaves) como entradas
mov al, 0        ; 5° línea faltante
out CB, al       ; configurar todos los conectores del puerto
                 ; B del PIO (luces) como salidas
in al, PA        ; leer y obtener el estado actual de las llaves
out PB, al       ; encender o apagar cada luz para que refleje y
                 ; se corresponda con el estado de su respectiva
                 ; llave (prendida o apagada respectivamente)
int 0
end
```

Ejercicio 2

a)

```
PIO         EQU 30H          ; llaves y luces
REG_PB      EQU PIO + 1      ; 31H → Puerto registro PB
                             ; (datos del puerto B del PIO: luces)
DATO_PB     EQU 11000011B    ; se encienden las primeras y las últimas
                             ; dos luces (0, 1, 6 y 7), mientras que se
                             ; apagan las demás (2 a 5)
REG_CB      EQU PIO + 3      ; 33H → Puerto registro CB
                             ; (control del puerto B del PIO: luces)
CTRL_PB     EQU 00000000B    ; todos los bits del puerto B son salidas

            ORG 2000H
            MOV AL, CTRL_PB
            OUT REG_CB, AL
            MOV AL, DATO_PB
            OUT REG_PB, AL
            INT 0
            END
```

b)

```
PIO         EQU 30H          ; llaves y luces
REG_PA      EQU PIO          ; 30H → Puerto registro PA
                             ; (datos del puerto A del PIO: llaves)
LLAVE7_ON   EQU 10000000B    ; llave 7 prendida
REG_CA      EQU PIO + 2      ; 32H → Puerto registro CA
                             ; (control del puerto A del PIO: llaves)
CTRL_PA     EQU 11111111B    ; todos los bits del puerto A son entradas

            ORG 1000H
MSJ_ON      DB "Llave prendida"
MSJ_OFF     DB "Llave apagada"
FIN_MSJ     DB ?

            ORG 2000H
            MOV AL, CTRL_PA
            OUT REG_CA, AL
            IN AL, REG_PA
            AND AL, LLAVE7_ON
            JZ APAGADA
```

```
            MOV BX, OFFSET MSJ_ON
            MOV AL, OFFSET MSJ_OFF - OFFSET MSJ_ON
            JMP IMPRIMIR
APAGADA:    MOV BX, OFFSET MSJ_OFF
            MOV AL, OFFSET FIN_MSJ - OFFSET MSJ_OFF
IMPRIMIR:   INT 7
            INT 0
            END
```

c)

```
PIO         EQU 30H          ; llaves y luces
REG_PA      EQU PIO          ; 30H → Puerto registro PA (llaves)
REG_PB      EQU PIO + 1      ; 31H → Puerto registro PB (luces)
DATO_PB     EQU 00000000B    ; se apagan todas las luces
REG_CA      EQU PIO + 2      ; 32H → Puerto registro CA (llaves)
CTRL_PA     EQU 11111111B    ; todos los bits del puerto A son entradas
REG_CB      EQU PIO + 3      ; 33H → Puerto registro CB (luces)
CTRL_PB     EQU 00000000B    ; todos los bits del puerto B son salidas

            ORG 2000H
            MOV AL, CTRL_PA
            OUT REG_CA, AL
            MOV AL, CTRL_PB
            OUT REG_CB, AL
            MOV AL, DATO_PB
            OUT REG_PB, AL
BUCLE:      IN AL, REG_PA
            OUT REG_PB, AL
            JMP BUCLE
            INT 0
            END
```

Parte 2: E/S con la Impresora mediante el PIO y el HAND-SHAKE

Ejercicio 1

a)

Tipo / Señal	Entrada	Salida
Data		✓
Strobe		✓
Busy	✓	

b)

Carácter enviado en la señal “Data”	Descripción de los eventos ocurridos
‘h’	Aunque la impresora está disponible (“busy” tiene un nivel bajo), en ningún momento se genera un flanco ascendente en “strobe”. Por lo tanto, la impresora ignora la línea de “data”: no se imprimirá el carácter ‘h’.
‘o’	La impresora está disponible (“busy” tiene un nivel bajo) y se genera un flanco ascendente en “strobe”. Por lo tanto, la impresora imprimirá el carácter ‘o’.
‘l’	Aunque se genera un flanco ascendente en “strobe”, la impresora está ocupada en ese instante de tiempo (“busy” tiene un nivel alto), probablemente imprimiendo el carácter ‘o’ previamente enviado. Por lo tanto, ignora la línea de “data”: no se imprimirá el carácter ‘l’.
‘a’	Se genera un flanco ascendente en “strobe” y la impresora está libre en ese instante de tiempo (“busy” tiene un nivel bajo). Por lo tanto, imprimirá el carácter ‘a’.
‘a’	Se genera un flanco ascendente en “strobe” y la impresora está libre en ese instante de tiempo (“busy” tiene un nivel bajo). Por lo tanto, imprimirá el carácter ‘a’.

En conclusión, la cadena impresa por la impresora será “oaa”, ignorándose los caracteres ‘h’ y ‘l’ de la cadena original “holaa”.

c)

Si no existiera la señal “strobe”, la impresora no tendría manera alguna de distinguir entre ambos caracteres ‘a’ a imprimir ya que, al tratarse en realidad del mismo carácter, el valor enviado por la señal “data” no cambia en ningún momento: el código ASCII transmitido permanece inalterable. En consecuencia, solo imprimiría un único carácter ‘a’ en lugar de los dos caracteres de la cadena “aa” que el usuario pretendía enviar.

En cambio, cada flanco ascendente generado en “strobe” permite indicarle a la impresora que necesariamente debe imprimir un nuevo carácter, cuya representación ASCII estará en la señal “data”. Esta indicación u orden resultará válida independientemente de si el carácter transmitido es idéntico o no a aquél previamente impreso, es decir, si hubo o no algún cambio en el valor de la señal “data”, la cual, tal como se afirmó y estableció, contiene el próximo carácter a imprimir.

Ejercicio 2

a)

```
PA          EQU 30h
CA          EQU 32h

            ORG 1000h
si          db "Ocupada"
no          db "Libre"

            ORG 2000h
            mov al, 00000001b
            ; 1° línea faltante: los bits 1 (STROBE) y 0 (BUSY) del
            ; puerto A del PIO (registro de estado y configuración de
            ; la impresora) se configuran como una salida y una entrada
            ; respectivamente (los demás resultan irrelevantes)
            out CA, al
            in al, PA
            and al, 00000001b
            ; 2° línea faltante: forzar a cero todos los bits leídos
            ; del registro de datos PA del puerto A del PIO, excepto el
            ; bit 0 (STROBE) ya que éste es aquél cuyo valor actual se
            ; debe comprobar
            jnz ocupada
            ; si STROBE es igual a uno, la impresora está ocupada, así
            ; que corresponde imprimir en pantalla el mensaje "Ocupada"
            mov bx, offset no
            ; 3° línea faltante: STROBE es igual a cero, por lo que la
            ; impresora está libre, así que corresponde imprimir en
            ; pantalla el mensaje "Libre"
            mov al, 5
            int 7
            jmp fin
ocupada:    mov bx, offset si
            mov al, 7
            int 7
            ; 4° línea faltante
fin:        int 0
            end
```

b)

```
PA          EQU 30h
CA          EQU 32h

            ORG 1000H
mensaje     db "Libre"

            ORG 2000H
            mov al, 00000001b
            out CA, al
consulta:   in al, PA
            and al, 00000001b
            jnz consulta
            ; mientras la impresora esté ocupada (STROBE sea igual a
            ; uno), se procederá a consultar repetidamente su estado
```

```
            ; hasta que se libere (STROBE sea igual a cero)
            mov bx, offset mensaje
            ; la impresora está libre, así que se imprimirá en pantalla
            ; el mensaje "Libre"
            mov al, 5
            int 7
fin:        int 0
            end
```

Ejercicio 4

a)

```
PIO         EQU 30H          ; impresora
REG_PA      EQU PIO          ; 30H → Puerto registro PA
                             ; (bits BUSY y STROBE)
REG_PB      EQU PIO + 1      ; 31H → Puerto registro PB
                             ; (carácter a imprimir)
DATO_PB     EQU 41H          ; carácter 'A'
REG_CA      EQU PIO + 2      ; 32H → Puerto registro CA
                             ; (bits BUSY y STROBE)
CTRL_PA     EQU 00000001B    ; los bits 1 (STROBE) y 0 (BUSY) del puerto
                             ; A son una salida y una entrada
                             ; respectivamente (los demás resultan
                             ; irrelevantes)
REG_CB      EQU PIO + 3      ; 33H → Puerto registro CB
                             ; (carácter a imprimir)
CTRL_PB     EQU 00000000B    ; todos los bits del puerto B son salidas
ES_BUSY     EQU 00000001B    ; impresora ocupada: bit 0 (BUSY) activo
STROBE_0    EQU 11111101B    ; desactivar validación (bit 1: STROBE) del
                             ; carácter transmitido a la impresora
STROBE_1    EQU 00000010B    ; activar validación (bit 1: STROBE) del
                             ; carácter transmitido a la impresora

            ORG 3000H
INI_PIO:    PUSH AX
            IN AL, REG_PA          ; se preserva el valor original del
                                   ; bit BUSY de la impresora
            AND AL, STROBE_0
            OUT REG_PA, AL
            MOV AL, CTRL_PA
            OUT REG_CA, AL
            MOV AL, CTRL_PB
            OUT REG_CB, AL
            POP AX
            RET

            ORG 3200H
POLL:       PUSH AX
BUCLE_P:    IN AL, REG_PA
            AND AL, ES_BUSY
            JNZ BUCLE_P
            POP AX
            RET
```

```
            ORG 3400H
FLANCO:     PUSH AX
            IN AL, REG_PA
            OR AL, STROBE_1
            OUT REG_PA, AL          ; para la impresora, la validación
                                    ; correspondiente al bit STROBE se
                                    ; activa por flanco ascendente
            IN AL, REG_PA
            AND AL, STROBE_0
            OUT REG_PA, AL
            POP AX
            RET

            ORG 3600H
IMP_CAR:    PUSH AX
            OUT REG_PB, AL
            CALL POLL
            CALL FLANCO
            POP AX
            RET

            ORG 2000H
            CALL INI_PIO
            MOV AL, DATO_PB
            CALL IMP_CAR
            INT 0
            END
```

b)

```
PIO         EQU 30H          ; impresora
REG_PA      EQU PIO          ; 30H → Puerto registro PA
                             ; (bits BUSY y STROBE)
REG_PB      EQU PIO + 1      ; 31H → Puerto registro PB
                             ; (carácter a imprimir)
REG_CA      EQU PIO + 2      ; 32H → Puerto registro CA
CTRL_PA     EQU 00000001B    ; los bits 1 (STROBE) y 0 (BUSY) del puerto
                             ; A son una salida y una entrada
                             ; respectivamente
REG_CB      EQU PIO + 3      ; 33H → Puerto registro CB
CTRL_PB     EQU 00000000B    ; todos los bits del puerto B son salidas
ES_BUSY     EQU 00000001B    ; impresora ocupada: bit 0 (BUSY) activo
STROBE_0    EQU 11111101B    ; desactivar validación (bit 1: STROBE) del
                             ; carácter transmitido a la impresora
STROBE_1    EQU 00000010B    ; activar validación (bit 1: STROBE) del
                             ; carácter transmitido a la impresora

            ORG 1000H
MSJ         DB "ORGANIZACIÓN Y ARQUITECTURA DE COMPUTADORAS"
FIN_MSJ     DB ?

            ORG 3000H
INI_PIO:    PUSH AX
            IN AL, REG_PA
            AND AL, STROBE_0
            OUT REG_PA, AL
```

```
            MOV AL, CTRL_PA
            OUT REG_CA, AL
            MOV AL, CTRL_PB
            OUT REG_CB, AL
            POP AX
            RET

            ORG 3200H
POLL:       PUSH AX
BUCLE_P:    IN AL, REG_PA
            AND AL, ES_BUSY
            JNZ BUCLE_P
            POP AX
            RET

            ORG 3400H
FLANCO:     PUSH AX
            IN AL, REG_PA
            OR AL, STROBE_1
            OUT REG_PA, AL
            IN AL, REG_PA
            AND AL, STROBE_0
            OUT REG_PA, AL
            POP AX
            RET

            ORG 3600H
IMP_CAR:    PUSH AX
            OUT REG_PB, AL
            CALL POLL
            CALL FLANCO
            POP AX
            RET

            ORG 3800H
IMP_CAD:    PUSH BX
            PUSH DX
BUCLE_CAD:  MOV AL, [BX]
            CALL IMP_CAR
            INC BX
            DEC DL
            JNZ BUCLE_CAD
            POP DX
            POP BX
            RET

            ORG 2000H
            CALL INI_PIO
            MOV BX, OFFSET MSJ
            MOV DL, OFFSET FIN_MSJ - OFFSET MSJ
            CALL IMP_CAD
            INT 0
            END
```

c)

```
PIO         EQU 30H          ; impresora
REG_PA      EQU PIO          ; 30H → Puerto registro PA
REG_PB      EQU PIO + 1      ; 31H → Puerto registro PB
REG_CA      EQU PIO + 2      ; 32H → Puerto registro CA
CTRL_PA     EQU 00000001B
REG_CB      EQU PIO + 3      ; 33H → Puerto registro CB
CTRL_PB     EQU 00000000B
ES_BUSY     EQU 00000001B    ; señal BUSY activa: impresora ocupada
STROBE_0    EQU 11111101B    ; desactivar señal STROBE (validación)
STROBE_1    EQU 00000010B    ; activar señal STROBE (validación)
CANT_IMP    EQU 5            ; cantidad de caracteres a imprimir

            ORG 1000H
MSJ_1       DB "A continuación, se le solicitará el ingreso de cinco ",
               "caracteres de a uno por vez.", 0AH, "Por favor, ",
               "ingrese el primer carácter:", 0AH
MSJ_2       DB "Por favor, ingrese el siguiente carácter:", 0AH
MSJ_3       DB "Fin del programa. Muchas gracias."
FIN_MSJ_3   DB ?

            ORG 1500H
CAR_IMP     DB ?

            ORG 3000H
INI_PIO:    PUSH AX
            IN AL, REG_PA
            AND AL, STROBE_0
            OUT REG_PA, AL
            MOV AL, CTRL_PA
            OUT REG_CA, AL
            MOV AL, CTRL_PB
            OUT REG_CB, AL
            POP AX
            RET

            ORG 3200H
POLL:       PUSH AX
BUCLE_P:    IN AL, REG_PA
            AND AL, ES_BUSY
            JNZ BUCLE_P
            POP AX
            RET

            ORG 3400H
FLANCO:     PUSH AX
            IN AL, REG_PA
            OR AL, STROBE_1
            OUT REG_PA, AL
            IN AL, REG_PA
            AND AL, STROBE_0
            OUT REG_PA, AL
            POP AX
            RET
```

```
            ORG 3600H
IMP_CAR:    PUSH AX
            OUT REG_PB, AL
            CALL POLL
            CALL FLANCO
            POP AX
            RET

            ORG 3800H
IMP_CAD:    PUSH BX
            PUSH CX
            PUSH DX
BUCLE_CAD:  PUSH BX
            MOV BX, CX
            INT 6
            MOV AL, [BX]
            POP BX
            CALL IMP_CAR
            DEC DH
            JZ FIN_CAD
            MOV AL, DL
            INT 7
            JMP BUCLE_CAD
FIN_CAD:    POP DX
            POP CX
            POP BX
            RET

            ORG 2000H
            CALL INI_PIO
            MOV BX, OFFSET MSJ_1
            MOV AL, OFFSET MSJ_2 - OFFSET MSJ_1
            INT 7
            MOV BX, OFFSET MSJ_2
            MOV CX, OFFSET CAR_IMP
            MOV DL, OFFSET MSJ_3 - OFFSET MSJ_2
            MOV DH, CANT_IMP
            CALL IMP_CAD
            MOV BX, OFFSET MSJ_3
            MOV AL, OFFSET FIN_MSJ_3 - OFFSET MSJ_3
            INT 7
            INT 0
            END
```

Ejercicio 5

a)

```
HAND        EQU 40H          ; dispositivo HAND-SHAKE
REG_DATO    EQU HAND         ; 40H → Puerto registro DATO: carácter
                             ;       a imprimir o ya impreso
REG_EST     EQU HAND + 1     ; 41H → Puerto registro EST: bits BUSY (0),
                             ;       STROBE (1) e INT (7)
EST_INI     EQU 01111111B    ; estado inicial del HAND-SHAKE: preservar
                             ; los valores originales de BUSY y STROBE
                             ; y desactivar INT
```

```
ES_BUSY    EQU 00000001B   ; señal BUSY activa: impresora ocupada

           ORG 1000H
MSJ        DB "INGENIERÍA E INFORMÁTICA"
FIN_MSJ    DB ?

           ORG 3000H
INI_HAND:  PUSH AX
           IN AL, REG_EST
           AND AL, EST_INI
           OUT REG_EST, AL
           POP AX
           RET

           ORG 3200H
POLL:      PUSH AX
BUCLE_P:   IN AL, REG_EST
           AND AL, ES_BUSY
           JNZ BUCLE_P
           POP AX
           RET

           ORG 3400H
IMP_CAR:   PUSH AX
           CALL POLL
           OUT REG_DATO, AL
           POP AX
           RET

           ORG 3600H
IMP_CAD:   PUSH BX
           PUSH DX
BUCLE_CAD: MOV AL, [BX]
           CALL IMP_CAR
           INC BX
           DEC DL
           JNZ BUCLE_CAD
           POP DX
           POP BX
           RET

           ORG 2000H
           CALL INI_HAND
           MOV BX, OFFSET MSJ
           MOV DL, OFFSET FIN_MSJ - OFFSET MSJ
           CALL IMP_CAD
           INT 0
           END
```

b)

Tal como se indica en la consigna del presente ejercicio, el HAND-SHAKE es un dispositivo diseñado específicamente para interactuar con la impresora mediante el protocolo Centronics. Por este motivo, a diferencia del PIO, no requiere generar manualmente el flanco ascendente en la señal STROBE para indicar a la impresora que debe imprimir un nuevo carácter. En lugar

de eso, al escribir un nuevo valor en su registro DATA, sea distinto o no a aquél almacenado hasta el momento en dicho registro, la mera operación de escritura provocará que el mismo HAND-SHAKE transmita el correspondiente flanco ascendente en STROBE hacia la impresora.
Al tratarse de un dispositivo con un propósito tan específico y exclusivo, su uso únicamente es compatible con el de la impresora. Ningún otro periférico de E/S puede conectarse a él: de hecho, se observa que, a diferencia del PIO, las llaves y luces, por ejemplo, no se encuentran disponibles en el simulador VonSim como una opción seleccionable para su utilización en forma conjunta con el HAND-SHAKE.

Parte 3: Entrada/Salida con dispositivos genéricos a través del PIO

Ejercicio 1

a)

```
PIO         EQU 30H          ; dispositivo PIO
REG_PB      EQU PIO + 1      ; 31H → Puerto registro PB
REG_CB      EQU PIO + 3      ; 33H → Puerto registro CB
CTRL_PB     EQU 00000000B    ; todos los bits del puerto B son salidas
VAL_CAB     EQU 0            ; valor de cabecera para cada envío
VAL_FIN     EQU 255          ; valor para indicar el fin de la
                             ; transmisión de la cadena

            ORG 1000H
CADENA      DB "UNLP"
FIN_CAD     DB ?

            ORG 3000H
INI_DISP:   PUSH AX
            MOV AL, CTRL_PB
            OUT REG_CB, AL
            POP AX
            RET

            ORG 3200H
SEND_CAB:   PUSH AX
            MOV AL, VAL_CAB
            OUT REG_PB, AL
            POP AX
            RET

            ORG 3400H
SEND_CAR:   PUSH AX
            CALL SEND_CAB
            OUT REG_PB, AL
            POP AX
            RET
```

```
            ORG 3600H
SEND_FIN:   PUSH AX
            MOV AL, VAL_FIN
            OUT REG_PB, AL
            POP AX
            RET

            ORG 3800H
SEND_CAD:   PUSH BX
            PUSH DX
            PUSH AX
BUCLE_CAD:  MOV AL, [BX]
            CALL SEND_CAR
            INC BX
            DEC DL
            JNZ BUCLE_CAD
            CALL SEND_FIN
            POP AX
            POP DX
            POP BX
            RET

            ORG 2000H
            CALL INI_DISP
            MOV BX, OFFSET CADENA
            MOV DL, OFFSET FIN_CAD - OFFSET CADENA
            CALL SEND_CAD
            INT 0
            END
```

b)

```
PIO         EQU 30H          ; dispositivo PIO
REG_PA      EQU PIO          ; 30H → Puerto registro PA
REG_PB      EQU PIO + 1      ; 31H → Puerto registro PB
REG_CA      EQU PIO + 2      ; 32H → Puerto registro CA
REG_CB      EQU PIO + 3      ; 33H → Puerto registro CB
CTRL_PA     EQU 11111111B    ; todos los bits del puerto A son entradas
CTRL_PB     EQU 00000000B    ; todos los bits del puerto B son salidas
VAL_LISTA   EQU 0FFH         ; CPU lista para recibir un nuevo carácter
VAL_FIN     EQU 0            ; fin de la transmisión de la cadena

            ORG 1000H
CADENA      DB ?

            ORG 3000H
INI_DISP:   PUSH AX
            MOV AL, CTRL_PA
            OUT REG_CA, AL
            MOV AL, CTRL_PB
            OUT REG_CB, AL
            POP AX
            RET
```

```
            ORG 3200H
LISTA:      PUSH AX
            MOV AL, VAL_LISTA
            OUT REG_PB, AL
            POP AX
            RET

            ORG 3400H
RECIB_CAR:  PUSH AX
            CALL LISTA
            IN AL, REG_PA
            MOV DL, AL
            POP AX
            RET

            ORG 3600H
RECIB_CAD:  PUSH BX
            PUSH DX
BUCLE_CAD:  CALL RECIB_CAR
            CMP DL, VAL_FIN
            JZ FIN_CAD
            MOV [BX], DL
            INC BX
            JMP BUCLE_CAD
FIN_CAD:    POP DX
            POP BX
            RET

            ORG 2000H
            CALL INI_DISP
            MOV BX, OFFSET CADENA
            CALL RECIB_CAD
            INT 0
            END
```

Parte 4: Ejercicios integradores o tipo parcial

Ejercicio 1

```
PIO             EQU 30H         ; llaves y luces
REG_PB          EQU PIO + 1     ; 31H → Puerto registro PB
DATO_PB         EQU 00000000B
REG_CB          EQU PIO + 3     ; 33H → Puerto registro CB
CTRL_PB         EQU 00000000B
LUZ_7           EQU 10000000B
LUZ_1           EQU 00000010B
LUZ_0           EQU 00000001B
DIR_IZQ         EQU 'I'
DIR_DER         EQU 'D'
N_ROT           EQU 1

                ORG 1000H
BYTE_LUCES      DB LUZ_1
```

```
                ORG 3000H
INI_DISP:       PUSH AX
                MOV AL, CTRL_PB
                OUT REG_CB, AL
                MOV AL, DATO_PB
                OUT REG_PB, AL
                POP AX
                RET

                ORG 3200H
ROTARIZQ:       PUSH AX
                PUSH BX
                MOV AL, [BX]
                ADD AL, AL
                ADC AL, 0
                MOV [BX], AL
                POP BX
                POP AX
                RET

                ORG 3400H
ROTARIZQ_N:     PUSH CX
BUCLE_IZQ:      CALL ROTARIZQ
                DEC CL
                JNZ BUCLE_IZQ
                POP CX
                RET

                ORG 3600H
ROTARDER_N:     PUSH DX
                PUSH CX
                MOV CL, 8
                SUB CL, DH
                CALL ROTARIZQ_N
                POP CX
                POP DX
                RET

                ORG 3800H
ROTACIONES:     PUSH BX
                PUSH DX
                PUSH AX
BUCLE_INF:      CMP DL, DIR_DER
                JZ SIG_DER
SIG_IZQ:        CALL ROTARIZQ
                JMP ENCENDER
SIG_DER:        CALL ROTARDER_N
ENCENDER:       MOV AL, [BX]
                OUT REG_PB, AL
                CMP AL, LUZ_0
                JZ ACT_DIR_IZQ
                CMP AL, LUZ_7
                JZ ACT_DIR_DER
                JMP BUCLE_INF
ACT_DIR_IZQ:    MOV DL, DIR_IZQ
```

```
                JMP BUCLE_INF
ACT_DIR_DER:    MOV DL, DIR_DER
                JMP BUCLE_INF
FIN_ROTACIONES: POP AX
                POP DX
                POP BX
                RET

                ORG 2000H
                CALL INI_DISP
                MOV BX, OFFSET BYTE_LUCES
                MOV DL, DIR_DER
                MOV DH, N_ROT
                CALL ROTACIONES
                HLT
                END
```

Ejercicio 2

```
PIO             EQU 30H         ; llaves y luces
REG_PA          EQU PIO         ; 30H → Puerto registro PA
REG_PB          EQU PIO + 1     ; 31H → Puerto registro PB
REG_CA          EQU PIO + 2     ; 32H → Puerto registro CA
REG_CB          EQU PIO + 3     ; 33H → Puerto registro CB
CTRL_PA         EQU 11111111B
CTRL_PB         EQU 00000000B
PATRON          EQU 01000101B
CORRECTO        EQU 11111111B

                ORG 1000H
MENSAJE         DB "GANASTE"
PATRON_MEM      DB PATRON

                ORG 3000H
INI_DISP:       PUSH AX
                MOV AL, CTRL_PA
                OUT REG_CA, AL
                MOV AL, CTRL_PB
                OUT REG_CB, AL
                POP AX
                RET

                ORG 3200H
VER_LLAVES:     IN AL, REG_PA
                XOR AL, [BX]
                NOT AL
                RET

                ORG 3400H
JUGAR:          PUSH BX
                PUSH AX
BUCLE_J:        CALL VER_LLAVES
                OUT REG_PB, AL
                CMP AL, CORRECTO
                JNZ BUCLE_J
                POP AX
```

```
POP BX
RET

ORG 2000H
CALL INI_DISP
MOV BX, OFFSET PATRON_MEM
CALL JUGAR
MOV BX, OFFSET MENSAJE
MOV AL, OFFSET PATRON_MEM - OFFSET MENSAJE
INT 7
INT 0
END
```