

Práctica 1-Semana 2- Presencial

1.4 Actividad Presencial: Programación en Ensamblador

1.

```
STACK_TOP    EQU    0x10002000

                AREA RESET, CODE
                DCD STACK_TOP
                DCD inicio
                ENTRY

inicio
                MOV     R0, #0             ; R0 = 0x00000000

llenar_1s
                ORR     R0, R0, #1         ; Insertar un '1' en el bit menos
significativo
                LSL     R0, R0, #1         ; Desplazar a la izquierda
                CMP     R0, #0xFFFFFFFF ; ¿ya está lleno de 1s?
                BNE     llenar_1s

vaciar_0s
                LSR     R0, R0, #1         ; Desplazar a la derecha
                CMP     R0, #0             ; ¿ya está a 0?
                BNE     vaciar_0s

bucle    B bucle                ; bucle infinito

                END
```

2.

```
STACK_TOP    EQU    0x10002000

                AREA RESET, CODE
                DCD STACK_TOP
                DCD inicio
                ENTRY

inicio
                LDR     R0, =Dato          ; Dirección del dato
                LDRB    R0, [R0]           ; Cargar Dato (0xA3)
                LDR     R1, =Bits          ; Dirección donde guardar los resultados
```

```

        PUSH    {R0, R1}          ; Pasar parámetros a la pila
        BL      Rutina            ; Llamada a la rutina
        ADD     SP, SP, #8        ; Restaurar puntero de pila
while    B       while            ; Bucle infinito

; -----
; Rutina
; -----
Rutina
        PUSH    {R0, R1, R2, R3, LR} ; Guardar contexto

        MOV     R2, #0            ; Contador de bits (0..7)
bucle_bits
        TST     R0, #1            ; Comprobar bit menos significativo
        MOVEQ   R3, #0x00        ; Si es 0 → guardar 0x00
        MOVNE   R3, #0xFF        ; Si es 1 → guardar 0xFF
        STRB    R3, [R1, R2]     ; Guardar en Bits + R2
        LSR     R0, R0, #1        ; Desplazar dato a la derecha
        ADD     R2, R2, #1        ; Incrementar índice
        CMP     R2, #8
        BNE     bucle_bits

        POP     {R0, R1, R2, R3, PC} ; Restaurar contexto y volver

; -----
; Datos
; -----
        AREA    Constantes, DATA, READONLY
Dato     DCB     0xA3             ; 10100011b

        AREA    HeaderData, DATA
Bits     SPACE   8                ; Espacio para almacenar los 8 bytes

        END

```