```
$include (c8051f020.inc)
          ; P3 & P2.0 & P2.1 are LED outputs
        dseg at 20h
Position: ds 1
old_button: ds 1
        cseg

        mov wdtcn,#0DEh          ; disable watchdog
        mov wdtcn,#0ADh
        mov xbr2,#40h            ; enable port output

        setb P2.7                ; Input button
        setb P2.6                ; Input button
        mov     Position, #04h   ; Starting position since Display incriments
        LCALL   Display



loop:   LCALL   DELAY
        LCALL   Check_buttons
        ANL     A, #11000000b    ; This takes the outputs off of A.
        CJNE    A, #90h, LEFT    ; If both buttons were pushed what to do
        SJMP    loop
LEFT:   CJNE    A, #80h, RIGHT   ; If only left button was pushed what to do
        inc     position
        LCALL   Display
        LJMP    Game_over
RIGHT:  CJNE    A, #40h, NONE    ; If only right button was pushed what to do
        dec     position
        LCALL   Display
        LJMP    Game_over
NONE:   sjmp    loop            ; Continue game if no button was pushed.

OVER:   SJMP    OVER             ; Ends program



; ------------- Checks the Buttons ------------
Check_buttons:
        mov     A, P2
        cpl     A                ; CPL inputs since active low.
        XCH     A, old_button    ; puts the value of the new
                                   buttons in storage and puts
                                   the value of the old buttons
                                   on the ACC
        XRL     A, old_button    ; If the buttons are the same
                                   change them to 0's
        ANL     A, old_button    ; If the buttons were different
                                   and they were pressed they stay.
        RET
; ------------- Display ------------
Display: ORL    P3, #0FFh
        ORL     P2, #03h
        mov     A, position
        LCALL   DISP_LED
        mov     A, position      ; Is this code needed, I don't
                                   think A will get changed while
                                   DISP_LED is ran.XXXXXXXXXX
        inc     A
        LCALL   DISP_LED
        ret
```

```
; ------------- Game Over ----------------
Game_over: mov     A, position
        CJNE           A, #00H, NINE ; If position = 0 then the
                                       game is over
        SJMP    OVER
NINE:   mov     A, position
        CJNE    A, #08h, loop     ; If position = 8 then the game
                                    is over, else continue.
        SJMP    OVER
; ------------- Display LED's ----------------
DISP_LED:
not_zero: CJNE  A, #00h, not_one  ; Compares accumulator with 0,
                                    if true it turns on the last
                                    light and ends the game.
        CLR     P3.0
        RET
not_one: CJNE   A, #01h, not_two  ; Compares accumulator with 1,
                                    if true it turns on the LED,
                                    if not it jumps to next bit
                                    if the accumulator bit is not 1.
        CLR     P3.1
        RET
not_two: CJNE   A, #02h, not_three
        CLR     P3.2
        RET
not_three: CJNE A, #03h, not_four
        CLR     P3.3
        RET
not_four: CJNE  A, #04h, not_five
        CLR     P3.4
        RET
not_five: CJNE  A, #05h, not_six
        CLR     P3.5
        RET
not_six: CJNE   A, #06h, not_seven
        CLR     P3.6
        RET
not_seven: CJNE A, #07h, not_eight
        CLR     P3.7
        RET
not_eight: CJNE A, #08h, not_nine
        CLR     P2.0
        RET
not_nine: CJNE A, #09h, not_one   ; if true it turns on the last
                                    light and ends the game.
        CLR     P2.1
        RET
; -------------Time Delay = 20 ms------------
DELAY:  mov     R2, #67         ; Load R2 with 67
otlp:   mov     R3, #200        ; Load R3 with 200,
                                ; 200 * 67 * 1.5 us = 20.1ms
inlp:   DJNZ    R3, inlp        ; Stay here till R3 = 0
        DJNZ    R2, otlp        ; Stay here till R2 = 0
        RET

        END
```