```
/* Program that displays a number on HEX0
 * that changes based on the key pressed:
 * 0 - Set to 0
 * 1 - Increment
 * 2 - Decrement
 * 3 - Clear (any key after that will set to 0)
 */
            .text
            .global _start

_start:     LDR     R6, =0xFF200020     // HEX3-HEX0 Address
            LDR     R7, =0xFF200050     // KEY Address
            MOV     R8, #BIT_CODES      // Address of BIT_CODES array
            MOV     R0, #0              // R0 will be the counter
MAIN:       LDR     R5, [R7]            // Read KEYs
            CMP     R5, #0
            BEQ     DISPLAY             // Check is no KEY has been pressed
            MOV     R4, R5              // Store KEY value when a key has been pressed
WAIT:       LDR     R5, [R7]            // Poll KEYs to see if the KEY has been released
            CMP     R5, #0
            BNE     WAIT                // Wait for KEY to be released

// Check which key has been pressed and act accordingly
ZERO:       CMP     R4, #0b0001         // Check if KEY0 is pressed
            BNE     INCREMENT
            MOV     R0, #0              // Set counter to 0
            B       DISPLAY
INCREMENT:  CMP     R4, #0b0010         // Check if KEY1 is pressed
            BNE     DECREMENT
            ADD     R0, #1              // Increment counter
            CMP     R0, #10             // Counter goes from 0 to 9, so wrap to 0 if = 10
            MOVEQ   R0, #0
            B       DISPLAY
DECREMENT:  CMP     R4, #0b0100         // Check if KEY2 is pressed
            BNE     CLEAR
            SUBS    R0, #1              // Decrement counter
            MOVMI   R0, #9              // Counter goes from 0 to 9, so wrap to 9 if =-1
            B       DISPLAY
CLEAR:      MOV     R1, #0              // If it gets to here KEY3 is definitely pressed
            STRB    R1, [R6]            // Set HEX to blank
CLEAR_WAIT: LDR     R5, [R7]            // Wait for any KEY to be pressed
            CMP     R5, #0
            BEQ     CLEAR_WAIT
            MOV     R4, #0b0001         // Set the KEY pressed to be "KEY0"
            B       WAIT                // Wait for the KEY to be released

DISPLAY:    LDRB    R1, [R8, +R0]       // Get digit to display
            STRB    R1, [R6]            // Display to HEX0
            B       MAIN                // Program infinitely counts/loops

BIT_CODES:  .byte   0b00111111, 0b00000110, 0b01011011, 0b01001111, 0b01100110
            .byte   0b01101101, 0b01111101, 0b00000111, 0b01111111, 0b01100111
            .skip   2               // pad with 2 bytes to maintain word alignment

```