Example #24 -- ARM data organization (arrays)

NUM = 6

.data
.balign 2
vector:
.short 21, -45, 96, 72, -33, 67
.global main
.text
.balign 4
main:
push {lr}
mov r4, #0 @ r4: sum of elements (zero to start)
mov r5, #0 @ r5: index into array (zero to start)
ldr r8, =vector @ r8: address of "vector"

loop:                 @ Compare current index to number of elements
    cmp r5, #NUM
    bge endloop
    lsl r6, r5, #1
    ldrsh r7, [r8,r6] @ r7: current array element
    add r4, r4, r7 @ r4: update sum of elements
    add r5, r5, #1 @ r5: update index into array
b loop
endloop:

ldr r0, =fmt @ 1st argument: address of format string
mov r1, r4 @ 2nd argument: sum of elements
bl printf @ Display the sum of the array elements

ldr r0, =vector @ 1st argument: starting address
add r1, #NUM @ 2nd argument: number of elements
bl display2 @ Display memory containing the array

pop {lr}
bx lr

fmt: .asciz "\nThe sum of the array elements:  %d\n"
.balign 4