.equ SWI\_Print\_Int, 0x6B

.equ SWI\_Exit, 0x11

.equ SWI\_Print\_Char, 0x00

.data

array:

.word 78, 23, 90, 42, 123

.text

.global \_start

\_start:

;; because we know the array is non-empty, this code

;; effectively does:

;;

;; int\* r2 = array; // r2 holds the address of an integer

;; int r3 = 5;

;; do {

;; println(\*r2);

;; r2++;

;; r3--;

;; } while (r3 != 0);

;;

;; r2: address of current element

;; r3: number of elements left to print

ldr r2, =array ; start on first element

mov r3, #5 ; 5 elements to print

loop\_begin:

;; print current element

mov r0, #1 ; write to standard output

ldr r1, [r2] ; read integer from array

swi SWI\_Print\_Int ; print the integer

;; print a newline

mov r0, #'\n

swi SWI\_Print\_Char

;; increment to the next array element

;; add 4 because words are 4 bytes long

add r2, r2, #4

;; decrement number of elements left

sub r3, r3, #1

;; if we have none left, we're done

cmp r3, #0

bne loop\_begin ; while (counter != 0)

;; exit the program

swi SWI\_Exit

.end