

② begin ! r1: direc. periférico
 .org 20400 ! r2: índice array
 .macro push arg1 ! r3: elem-
 add %r14, -4, %r14 actual
 st arg1, %r14

.end macro

.macro pop arg1
 ld %r14, arg1
 add %r14, 4, %r14

.end macro

len: equ 400
 direccion_perif: equ

sethi %hi(direccion_perif), %r1

or %lo(direccion_perif), %r1, %r1

add %r0, len, %r2

for: andcc %r2, %r2, %r0

⊕ add %r0, %r15, %r16

be done

push %r1

call leef

pop %r3

andcc %r3, %r3, %r0

bneg cargar

ba for

cargar: st %r3, %r2 + [Array]

%r2, -4, %r2

ba for


```

leer: pop r5
      ld r6, r5
      push r6
      jmp r15, 4, r0
done: jmp r15, 4, r0
Array: dwb 100
      .end

```

! la dirección de proceso invocante previamente guardada.

El programa define los macros push y pop, guarda la dirección del periférico usando xthi por el tamaño de la dirección, define las constantes len(100 * 4 = 400) y direc_periferico, guarda la longitud en un registro, y en un loop verifica si el valor leído del periférico es negativo (usando branch if negative). Para cargar los valores al arreglo (el cual reservó un espacio de memoria con "dwb 100"), utiliza la posición del arreglo (es una constante) y le suma el índice actual (que solo se actualiza si el valor leído es negativo).