```
.data


ar:  .word -1, -10, 482, -280, 41, 60, -11, -20, 211, 120
 .word 14, 70, 91, 409, 21, 40, 81, 90, 901, 110
 .word 1, 10, 11, 10, 11, 10, 111, 110, 11, 1230
 .word 13, 120, 321, -40, -61, 550, 411, -340, 13, 50
 .word 14, 50, 41, -560, 300, -200, 1, 0, 1, 40

# $8 -- a flag, 1 if the algorithm is done
# $9 -- an offset to the correct element of the array
# $10 -- address of the element to compare
# $11 -- the array element for comparison
# $12 -- the neighbor of the array element for comparison
# $14 -- base address of array ar

.text
.globl main
main: la $14, ar
loop:  li $8, 1 # flag = true
 li $9, 0
 for:  add $10, $14, $9
  lw $11, ($10) # load element
  lw $12, 4($10) # load next element
  sub $13, $11, $12
  blez $13, noswap # if they are in order, don’t swap
  li $8, 0
  sw $11, 4($10) # swap elements
  sw $12, ($10)
 noswap: add $9, $9, 4
  sub $13, $9, 196 # see if end of the array reached
 bltz $13,for
beq $8, $0, loop # loop until done
li $v0, 10
syscall
```