

```

## minmax.asm - print min, max of array elements.
##
## Assumes the array has at least two elements (a[0]
## and a[1]). It initializes both min and max to a[0]
## and then goes through the loop count-1 times.
## This program will use pointers.
##
## t0 - points to array elements in turn
## t1 - contains count of elements
## t2 - contains min
## t3 - contains max
## t4 - each word from array in turn

#####
#
# text segment
#
#####

.text
.globl main
main:
    la $t0,array # $t0 will to elements
    lw $t1,count # exit loop when $t1 is 0
    lw $t2,($t0) # initialize both min ($t2)
    lw $t3,($t0) # and max ($t3) to a[0]
    add $t0,$t0,4 # pointer to start at a[1]
    add $t1,$t1,-1 # and go round count-1 times

loop: lw $t4,($t0) # load next word from array
      bge $t4,$t2,notMin # skip if a[i] >= min
      move $t2,$t4 # copy a[i] to min

notMin: ble $t4,$t3,notMax # skip if a[i] <= max
        move $t3,$t4 # copy a[i] to max

notMax: add $t1,$t1,-1 # decrement counter
        add $t0,$t0,4 # increment pointer by word
        bnez $t1,loop # and continue if counter>0

    la $a0,ans1
    li $v0,4
    syscall # print "min = "

    move $a0,$t2
    li $v0,1
    syscall # print min

    la $a0,ans2
    li $v0,4
    syscall # print "\nmax = "

```

```

move $a0,$t3
li $v0,1
syscall # print max

la $a0,end1 # system call to print
li $v0,4 # out a newline
syscall

li $v0, 10
syscall # au revoir

#####
#
# data segment
#
#####

.data

array: .word 3,4,2,6,12,7,18,26,2,14,19,7,8,12,13
count: .word 15
end1: .asciiz "\n"
ans1: .asciiz "min = "
ans2: .asciiz "\nmax = "

##
## end of file minmax.asm

```