```
#####################################################################
#    MIPS Assembly program that shows how to implement an array     #
#####################################################################

    .data

mintext: .asciiz "minimum sum: "
indextext: .asciiz "  at index: "
# an array of 10 coordinates (x,y,z)
array: .word
    1, 10, 18,        # 0
    2, 2, 20,         # 1
    13, 13, 1,        # 2
    20, 20, 100,      # 3
    8, 9, 10,         # 4
    11, 12, 1,        # 5
    20, 1, 2,         # 6
    18, 8, 8,         # 7
    9, 9, 3,          # 8
    10, 9, 5          # 9

    .text

  li $t8, 999999   # t8 stores minsum
  li $t0, 0
  li $t1, 10
startloop:
  beq $t0, $t1, exitloop
  la $a0, array
  mul $t3, $t0, 12
  add $a0, $a0, $t3   # a0 = array + 3*4*i
  lw $t4, 0($a0)       # x_i
  lw $t5, 4($a0)       # y_i
  add $t4, $t4, $t5
  lw $t5, 8($a0)       # z_i
  add $t4, $t4, $t5   # t4 = x_i + y_i + z_i
  bgt $t4, $t8, continue  #jump if sum is larger than old minimum sum
     #if not, then new minimum sum found
  move $t8, $t4  # save new minimum
  move $t7, $t0  # save index
continue:
  addi $t0, $t0, 1    # increment i
  j startloop
exitloop:
  li $v0, 4
  la $a0, mintext
  syscall
  li $v0, 1
  move $a0, $t8
  syscall
  li $v0, 4
  la $a0, indextext
  syscall
  li $v0, 1
  move $a0, $t7
  syscall
#terminate program
  li $v0, 10
  syscall
```

```
##########################################################################
#    MIPS Assembly program that shows how to implement a struct     #
##########################################################################

    .data

nameprompt: .asciiz "name:"
ageprompt: .asciiz "age:"
genderprompt: .asciiz "gender:"
cityprompt: .asciiz "City:"
mystruct: .word 0:129
    # name:   256 chars ASCII = 64 words
    # age:    1 byte = 1 word
    # gender: 1 char = 1 word
    # city:   256 chars ASCII = 64 words
    #--------------------------+
    #            130 words =  520 bytes

.text

    li $v0, 4
    la $a0, nameprompt
    syscall
    la $a0, mystruct
    li $a1, 256
    li $v0, 8
    syscall   # read string. $a0 = string address, $a1 = max length
    li $v0, 4
    la $a0, ageprompt
    syscall
    li $v0, 5
    syscall   # read int into $v1
    la $a0, mystruct
    addi $a0, $a0, 256  # we have to calculate where the age int is
                        #  in the struct
    sw $v0, 0($a0)
    li $v0, 4
    la $a0, genderprompt
    syscall
    la $a0, mystruct
    addi $a0, $a0, 260 # we have to calculate where the gender byte is
                        #  in the struct
    li $a1, 10 # it reads max 9 chars. Note: It may overwrite the
               #  adjoining city string!
    li $v0, 8
    syscall   # read string
    li $v0, 4
    la $a0, cityprompt
    syscall
    la $a0, mystruct
    addi $a0, $a0, 264 # we have to calculate where the city string is
                        #  in the struct
    li $a1, 256
    li $v0, 8
    syscall   # read string. $a0 = string address, $a1 = max length

#terminate program
    li $v0, 10
    syscall
```