```
.data
      A: .word 10, 11, -4, 6, 1,  4, 42, -11, 0, 3
      Msg1: .asciiz "Please enter a number: "
      Msg2: .asciiz "The sum is: "
.text
main:
      li $t0, 0                 # initialize counter
      li $t1, 10                # set the counter limit to 10  while (i ≠ 10)
      la $t2, A    pointer $t2 = &A; # set pointer to start of array
      li $t3, 0                 # initialize sum
AddNext: next   offset 0
      lw $t4, 0($t2) pointer $t2  # get next number from array
      add $t3, $t3, $t4 next    # add to sum
              sum
      addi $t2, $t2, 4 pointer += 4; # move the pointer
      addi $t0, $t0, 1  i++;    # increment counter
      bne $t0, $t1, AddNext     # repeat if not done
      la $a0, Msg2              # Output message
      li $v0, 4
      syscall
      li $v0, 1                 # next lines print sum
      move $a0, $t3
      syscall
      li $v0, 10                # next two lines STOP
      syscall
```

```
.data
      A: .space 40
      Msg1: .asciiz "Please enter a number: "
      Msg2: .asciiz "The sum is: "
.text
main:
      li $t0, 0                 # initialize counter
      li $t1, 10                # set the limit to 10
      la $t2, A                 # set pointer to the start of A
###### READING PART
ReadNext:
      la $a0, Msg1              # Input prompt
      li $v0, 4
      syscall
      li $v0, 5                 # read the next number
      syscall
      sw $v0, 0($t2)            # store the number
      addi $t2, $t2, 4          # move the pointer
      addi $t0, $t0, 1          # increment counter
      bne $t0, $t1, ReadNext    # repeat if not done
######  ADDING PART
      li $t0, 0                 # reinitialize counter
      la $t2, A                 # reset pointer to start of array
      li $t3, 0                 # initialize sum
AddNext:
      lw $t4, 0($t2)            # get next number from array
      add $t3, $t3, $t4         # add to sum
      addi $t2, $t2, 4          # move the pointer
      addi $t0, $t0, 1          # increment counter
      bne $t0, $t1, AddNext     # repeat if not done
######
      la $a0, Msg2              # Output message
      li $v0, 4
      syscall
      li $v0, 1                 # next lines print sum
      move $a0, $t3
      syscall
      li $v0, 10                # next two lines STOP
      syscall
```