```
##
##  This program is written to find out the number of vowels and the number of non-vowels in a given string
##  and use procedure call to check if a letter is vowel or not.
##
##      - It will have initilised string name "example" that
##        holds the given example
##      - then call a procedure, vowelp to check if it is a vowel.
##      - It will return a value to the calling function,
##        0 means consonent, and 1 means vowel,
##      - and then print number of vowels and consonents.
##
##      v0 - used for syscalls also holds 0 or 1 after the call to "vowelp"
##      t0 - holds example
##      t1 - vowels counter
##      t2 - consonents counter
##      t3 - holds the returned value from the procedure call
##      a0 - holds strings also to give letter to vowelp
##
##

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

    .text
    .globl __start
__start:                # execution starts here


    la $t0, example     # loading address of string
    li $t1, 0           # vowel count
    li $t2, 0           # consonent count

loop:
    lb $a0,($t0)        # loading single character from example
    beqz $a0, End       # if end of string then jimp to End
    addi $t0,$t0, 1     # or else move character by character in example
    jal vowelp          # procedure call
    move $t3,$v0        # saving value temporarily
    bnez $t3, plusVowel # If return value is 1 than the character is vowel
    j plusConso         # else jump to plusConso

plusConso:
    add $t2,1           #increment consonent count
    j loop              #reenter loop

plusVowel:
    addi $t1,$t1,1      #increment vowel count
    j loop              #reenter loop

End:
    la $a0, ans1        #syscall to print
    li $v0, 4           #result
    syscall

    move $a0,$t1
    li $v0, 1           #print number of vowels
    syscall

    la $a0, endl        #syscall to print out
    li $v0, 4           #a new line
    syscall

    la $a0, ans2        #syscall to print out
    li $v0, 4           #result
    syscall
    move $a0,$t2
    li $v0,1            #print number of consonents
    syscall

    la $a0,endl         #syscall to print out
    li $v0,4            #a new line
    syscall

    li $v0,10
    syscall             # Bye!

##############################################################
#                                                            #
#                      FUNCTION: vowelp                      #
#                                                            #
#   - checks for vowels if they are vowel or not             #
#                                                            #
##############################################################

vowelp:

    li $v0,0
    beq $a0,'A',yes
    beq $a0,'a',yes
    beq $a0,'E',yes
    beq $a0,'e',yes
    beq $a0,'I',yes
    beq $a0,'i',yes
    beq $a0,'O',yes
    beq $a0,'o',yes
    beq $a0,'U',yes
    beq $a0,'u',yes
    jr $ra
yes:
    addi $v0,$v0, 1     #If its vowel than return 1
    jr $ra


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

        .data
    example:    .asciiz "I am going to run this program with the real hardware in two weeks."
    ans1:       .asciiz "Vowels are : "
    endl:       .asciiz "\n"
    ans2:       .asciiz "Consonents are : "

#   end of file num-vowel.s
```