The following program uses a system call (syscall) instruction to read integers one at a time until a zero is typed. The program computes and prints the sum of all the integers typed.

```assembly
.data
prompt: .asciiz "Type an integer: 
sum_str: .asciiz "Sum = 
nl: .asciiz "\n" # String with newline character.

.text
.globl main

#Initialize.
main: move $16, $0 #$16 will contain the sum.

#Prompt the user for the next integer.
loop: la $a0, prompt #Start addr of prompt string.
    li $v0, 4 #print_string command.
syscall

#Read the integer using syscall.
    li $v0, 5 # 5 represents read_int command.
syscall # The integer is read into $v0.

#Check whether the integer is zero. If so print the sum.
#Otherwise, update the sum in $16.
    beq $v0, $0, print
    add $16, $16, $v0
    j loop
```

(over)
#Print sum and stop.

print:  la $a0, sum_str  #Start address of output string.
li $v0, 4
syscall

#Output the sum followed by the newline character.

move $a0, $16  #The value to be output must be in $a0.
li $v0, 1     # 1 represents print_int command.
syscall       # The integer in $a0 will be printed.
l $a0, nl     #Start address of string with '\n'.
l $v0, 4
syscall

li $v0, 10    #exit command.
syscall