```
        .data
msg:    .asciiz "Enter your name: "
amn:    .asciiz "Enter your AM: "
st1:    .asciiz "\nYour name is "
st2:    .asciiz "and your AM  is "
nam:    .space 255


        .text
        .globl main
main:   li $v0, 4       # syscall 4 (print_str)
        la $a0, msg     # argument: string
        syscall         # print the string

    la $a0, nam # read a string
    li $a1, 255     # of 255 bytes maximum
    li $v0, 8   # syscall 8 (read_str)
    syscall     # read the string

        li $v0, 4       # syscall 4 (print_str)
        la $a0, amn     # argument: string
        syscall         # print the string

    li $v0, 5       # syscall 4 (read_int)
        syscall         # read the integer
    add $t2, $zero, $v0
#   move $t2, $v0   #  ... another approach

        li $v0, 4       # syscall 4 (print_str)
        la $a0, st1     # argument: string
        syscall         # print the string

        li $v0, 4       # syscall 4 (print_str)
        la $a0, nam     # argument: string
        syscall         # print the string

        li $v0, 4       # syscall 4 (print_str)
        la $a0, st2     # argument: string
        syscall         # print the string

    add $a0, $zero, $t2
#   move $a0, $t2   # argument: integer ... another approach
        li $v0, 1       # syscall 1 (print_int)
        syscall         # print the integer

        jr $ra          # retrun to caller
```