```
#Noah del Angel, CS 2318 - 002, Assignment 2 Part 1 A
#Problem Description:
#Prompt the user to enter an integer, read the user's input, and display a
#   labeled output about the user's input.
#Prompt the user to enter a string of up to 50 characters long, read the
#   user's input, and display a labeled output about the user's input.
#   -String entered MUST be stored in separate storage space (i.e., not
#      overwriting the storage space used by prompt and label strings)
#      allocated just enough (no more, no less) to accommodate up to the
#      maximum number of characters indicated.
#Prompt the user to enter a character, read the user's input, and display a
#   labeled output about the user's input
#Note that it involves a character, NOT a one-charater string. You will get
#   no credits if you do it using a one-character string even though the
#   same output is obtained.
                    .data
intIPrmpt:          .asciiz "Enter a integer: "
intILabel:          .asciiz "The integer is "
strngIPrmpt:        .asciiz "Enter a string: "
strngILabel:        .asciiz "The string is "
strngI:             .space 51
charIPrmpt:         .asciiz "Please enter a character: "
charILabel:         .asciiz "The character is: "


                    .text
                    .globl main

main:
                    #Print int prompt
                    li $v0, 4
                    la $a0, intIPrmpt
                    syscall

                    #Get input
                    li $v0, 5
                    syscall
                    move $t0, $v0

                    #Print int message
                    li $v0, 4
                    la $a0, intILabel
                    syscall

                    #Print int input
                    li $v0, 1
                    move $a0, $t0
                    syscall



                    #Print a new line
```

```
li $v0, 11
li $a0, '\n'
syscall
li $v0, 11
li $a0, '\n'
syscall


#Print string prompt
li $v0, 4
la $a0, strngIPrmpt
syscall

#Get input
li $v0, 8
la $a0, strngI
li $a1, 51
syscall

#Print string label
li $v0, 4
la $a0, strngILabel
syscall

#Print string input
li $v0, 4
la $a0, strngI
syscall


#Print a new line
li $v0, 11
li $a0, '\n'
syscall
li $v0, 11
li $a0, '\n'
syscall


#Print char prompt
li $v0, 4
la $a0, charIPrmpt
syscall

#Get input
li $v0, 12
syscall
move $t0, $v0

#Print a new line
li $v0, 11
li $a0, '\n'
```

```
syscall

#Print char label
li $v0, 4
la $a0, charILabel
syscall

#Print char input
li $v0, 11
move $a0, $t0
syscall
```