```
1  ###############################################################################
2  # <Ben Hunt>, CS 2318-253, Assignment 2 Part 1 Program A
3  ###############################################################################
4  # Simple program to ask for user input for Ints, Strings, and Chars
5  # Outputs the entered values to the console.
6  ############################## data segment ###################################
7
8  .data
9
10 promptInt:  .asciiz "Enter an Integer: "
11 promptString: .asciiz "Enter a String: "
12 promptChar: .asciiz "Enter a Char: "
13 outputMsg: .asciiz "User entered: "
14 newLine: .asciiz "\n"
15 string: .space 41
16
17 ############################## code segment ###################################
18
19
20
21 .text
22
23 main:  .globl main
24
25      la $a0, promptInt   #adress of prompt is $a0
26      li $v0, 4           #Put value 4 into $v0; code 4 is print string
27      syscall             #used to read or print values or strings from input/output
window, and indicate program end
28
29      li $v0, 5        #Read in integer; code 5 is read integer
30      syscall          #used to read or print values or strings from input/output
window, and indicate program end
31
32      move $t0, $v0       # $t0 = $v0
33      la $a0, outputMsg   # load address of output into $a0
34      li $v0, 4           #Print string
35      syscall
36
37      move $a0, $t0   # $a0 = $t0 Note: [$v0, 1 requires argument $a0 where $a0 =
integer to be printed.
38      li $v0, 1       #Print integer with code 1
39      syscall
40
41
42      la $a0, newLine
43      li $v0, 4               #print string code is 4
44      syscall
45
46      la $a0, promptString    #prompt user to enter string, code is 4
47      li $v0, 4
48      syscall
49
50      la $a0, string  #put address of string into $a0
51      li $a1, 41      #Specify buffer size into $a1
52      li $v0, 8       #Read string is code 8
53      syscall
54
55      move $t1, $a0       # $t1 = $a0
56      la $a0, outputMsg   # $a0 = address of outputMsg
57      li $v0, 4           # Print String code 4
58      syscall
59
60      move $a0, $t1    # $a0 = $t1
61      li $v0, 4        # Print string
62      syscall
63
```

```
64      la $a0, promptChar  # $a0 = address of promptChar string
65      li $v0, 4           #Print string w/ code 4
66      syscall
67
68      li $v0, 12      # Read in character w/ code 12
69      syscall
70      move $t2, $v0   # $t2 = $v0 (save that read in char to $t2)
71
72      la $a0, newLine # $a0 = adress of newLine
73      li $v0, 4       #Prints out newLine
74      syscall
75
76      la $a0, outputMsg   # $a0 = address of outputMsg
77      li $v0, 4           # print out outputMsg
78      syscall
79
80      move $a0, $t2   #put that saved char value into $a0; $a0 = $t2
81      li $v0, 11      #print the char with code 11
82      syscall
83
84
85      li $v0, 10
86      syscall
87
88
89
```