

```

1  .data
2  str1:    .asciiz "Enter an integer value: "
3  str2:    .asciiz "You entered "
4
5  .globl   main
6  .text
7  main:
8      li    $v0, 4      # service code for print string
9      la    $a0, str1   # load address of str1 into $a0
10     syscall          # print str1 string
11     li    $v0, 5      # service code for read integer
12     syscall          # read integer input into $v0
13     move  $s0, $v0     # save input value in $s0
14     li    $v0, 4      # service code for print string
15     la    $a0, str2   # load address of str2 into $a0
16     syscall          # print str2 string
17     li    $v0, 1      # service code to print integer
18     move  $a0, $s0     # copy input value
19     syscall          # print integer
20     li    $v0, 10     # service code to exit program
21     syscall          # exit program

```

Figure 2.4: MIPS Program that uses System Calls

## 2.6 Exercise

1. Modify the program shown in Figure 2.4. Ask the user to enter an integer value, and then print the result of doubling that number. Use the **add** instruction.
2. Modify again the program shown in Figure 2.4. Ask the user whether he wants to repeat the program: "\nRepeat [y/n]? ". Use service code 12 to read a character and the branch instruction to repeat the main function if the user input is character 'y'.
3. Write a MIPS program that asks the user to input his name and then prints "Hello ", followed by the name entered by the user.
4. Write a MIPS program that executes the statement:  $s = (a + b) - (c + 101)$ , where  $a$ ,  $b$ , and  $c$  are user provided integer inputs, and  $s$  is computed and printed as an output. Answer the following:
  - a. Suppose the user enters  $a = 5$ ,  $b = 10$ , and  $c = -30$ , what is the expected value of  $s$ ?
  - b. Which instruction in your program computed the value of  $s$  and which register is used?
  - c. What is the address of this instruction in memory?
  - d. Put a breakpoint at this instruction and write the value of the register used for computing  $s$  in decimal and hexadecimal.
5. Write a MIPS program that inputs two integer values. The program should output **equal** if the two integers are equal. Otherwise, it should output **not equal**. Use the branch instruction to check for equality.