Program 4

Snippet:

```
#calculate factorial of a number
.data
string1: .asciiz "\nEnter a number of your choice:\n"
string2: .asciiz "\nThe factorial is:\n"
string3: .asciiz "\nThe number entered is less than 0\n"
.text
la $a0,string1
li $v0,4
syscall
#accept number from the user
li $v0,5
syscall
move $s0,$v0 #move the number into s0 register

#check condition
bltz $s0, error
move $a0, $s0 #pass the number as an argument to function

#call factorial
jal factorial
move $s1, $v0 #move value of $r0 into $s1

#print correct result
la $a0, string2
li $v0, 4 #to print strings
syscall

la $v0, 1
move $a0, $s1
syscall


# end program
li $v0, 10
syscall

error:
#print the wrong message if input is less than 0
la $a0, string3
li $v0, 4 #to print strings
syscall

# end program
li $v0, 10
syscall
```

```

factorial:
addi $sp, $sp, -8 #adjust stack for 2 items
sw $ra, 4($sp) #save return address
sw $a0, 0($sp) #save argument
slti $t0, $a0, 1 #test for n < 1
beq $t0, $zero, L1
addi $v0, $zero, 1 #if so, result is 1
addi $sp, $sp, 8 #pop 2 items from stack
jr $ra #and return

L1:
add $a0, $a0, -1 # else decrement n
jal factorial #recursive call
lw $a0, 0($sp) #restore original n
lw $ra, 4($sp) #and return address
addi $sp, $sp, 8 #pop 2 items from stack
mul $v0, $a0, $v0 #multiply to get result
jr $ra #and return
```

Output:

```
Enter a number of your choice:
**** user input : -3

The number entered is less than 0

-- program is finished running --
```

```
Enter a number of your choice:
**** user input : 5

The factorial is:
120
-- program is finished running --
```

Script:

```
#calculate factorial of a number
.data
string1: .asciiz "\nEnter a number of your choice:\n"
string2: .asciiz "\nThe factorial is:\n"
string3: .asciiz "\nThe number entered is less than 0\n"
.text
la $a0,string1
li $v0,4
syscall
#accept number from the user
li $v0,5
syscall
move $s0,$v0 #move the number into s0 register

#check condition
bltz $s0, error
move $a0, $s0 #pass the number as an argument to function

#call factorial
jal factorial
move $s1, $v0 #move value of $v0 into $s1

#print correct result
la $a0, string2
li $v0, 4 #to print strings
syscall

la $v0, 1
```

```
move $a0, $s1
syscall



# end program
li $v0, 10
syscall

error:
#print the wrong message if input is less than 0
la $a0, string3
li $v0, 4 #to print strings
syscall

# end program
li $v0, 10
syscall

factorial:
addi $sp, $sp, -8 #adjust stack for 2 items
sw $ra, 4($sp) #save return address
sw $a0, 0($sp) #save argument
slti $t0, $a0, 1 #test for n < 1
beq $t0, $zero, L1
addi $v0, $zero, 1 #if so, result is 1
addi $sp, $sp, 8 #pop 2 items from stack
jr $ra #and return

L1:
```

```
add $a0, $a0, -1 # else decrement n
jal factorial #recursive call
lw $a0, 0($sp) #restore original n
lw $ra, 4($sp) #and return address
addi $sp, $sp, 8 #pop 2 items from stack
mul $v0, $a0, $v0 #multiply to get result
jr $ra #and return
```