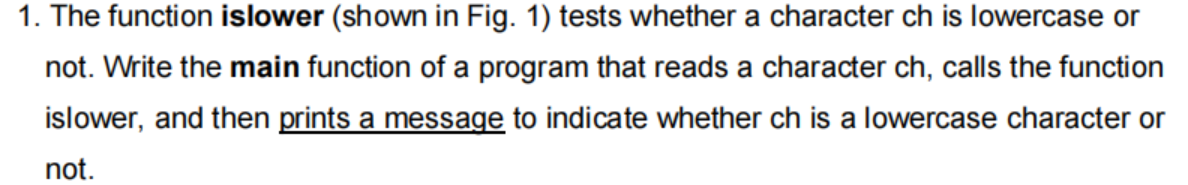
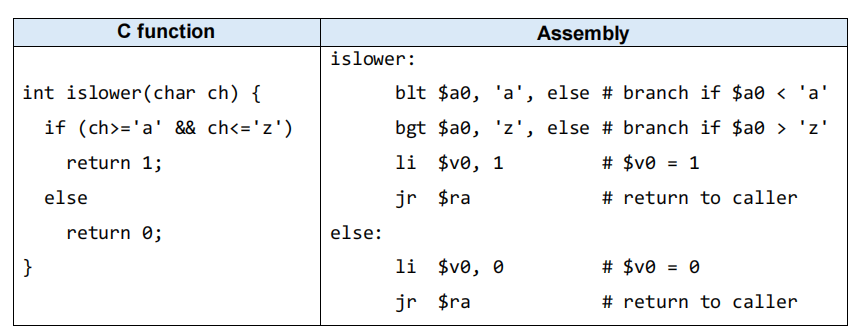
## LAB #06 Report

|  |  |  |
| --- | --- | --- |
| Name:黄家睿 | ID number:202283890036 | Major:IOT |

Task 1:





**Solution**

**Code：**

.data

str1: .asciiz "The char is lowercase"

str2: .asciiz "The char is upcase"

prompt:.asciiz "Enter a charater"

.text

main:

# Print the prompt message

li $v0, 4 # syscall code for print string

la $a0, prompt # address of the prompt message

syscall

# Read a character

li $v0, 12 # syscall code for read character

syscall

move $a0, $v0 # move the character to $a0

jal islower

beq $v0,1,print\_lowercase

print\_not\_lowercase:

li $v0, 4 # syscall code for print string

la $a0, str2 # address of the not lowercase message

syscall

j end\_program

print\_lowercase:

li $v0, 4 # syscall code for print string

la $a0, str1 # address of the lowercase message

syscall

end\_program:

li $v0, 10 # syscall code for exit

syscall

islower:

blt $a0,'a',else

bgt $a0,'z',else

li $v0,1

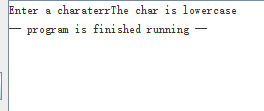
jr $ra

else:

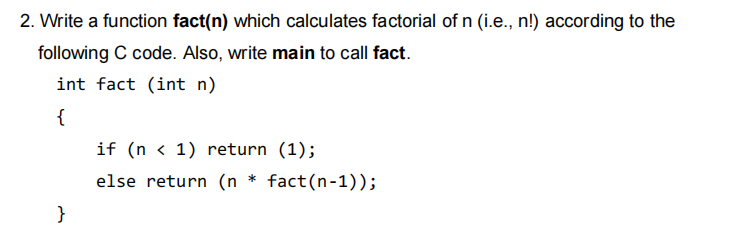
li $v0,0

jr $ra

**output：**



### Task 2:



**Solution:**

**Code:**

.data

prompt: .asciiz "Enter a number: "

result\_msg: .asciiz "The factorial is: "

newline: .asciiz "\n"

.text

.globl main

main:

# Print the prompt message

li $v0, 4 # syscall code for print string

la $a0, prompt # address of the prompt message

syscall

# Read an integer from the user

li $v0, 5 # syscall code for read integer

syscall

move $a0, $v0 # move the input number to $a0

# Call the fact function

jal fact

# Print the result message

li $v0, 4 # syscall code for print string

la $a0, result\_msg # address of the result message

syscall

# Print the factorial result

move $a0, $v0 # move the result from $v0 to $a0

li $v0, 1 # syscall code for print integer

syscall

# Print a newline

li $v0, 4 # syscall code for print string

la $a0, newline # address of the newline character

syscall

# Exit the program

li $v0, 10 # syscall code for exit

syscall

fact:

addi $sp, $sp, -8 # Create space on the stack

sw $ra, 4($sp) # Save return address

sw $a0, 0($sp) # Save argument n

bge $a0, 1, recurse # If n >= 1, recurse

li $v0, 1 # Base case: fact(0) = 1 or fact(n) for n < 1 = 1

j end\_fact # Return

recurse:

addi $a0, $a0, -1 # Calculate fact(n-1)

jal fact # Recursive call

lw $a0, 0($sp) # Restore argument n

mul $v0, $a0, $v0 # Multiply n \* fact(n-1)

end\_fact:

lw $ra, 4($sp) # Restore return address

addi $sp, $sp, 8 # Restore stack

jr $ra # Return

**Output:**

