LAB #04 Report

Name:黄家睿 ID number:202283890036 Major:IOT

Task1:

1. Write a program that asks the user to enter an integer and then displays the number of 1's in the binary representation of that integer. For example, if the user enters 9, then the program should display 2.

Solution:

Code:

```
.data

str1: .asciiz "Please enter a integer: "
str2: .asciiz "Number of 1's in binary representation: "

.globl main
.text
main:
        #input the integer
        li $v0, 4
        la $a0, str1
        syscall

        #read the input integer
        li $v0,5
        syscall
        move $t0,$v0     #save inout integer into $t0

        li $t1,0         #intialize a counter for the number of 1 is zero
```

```
check_loop:

        andi $t2, $t0, 1
        beqz $t2,shift_loop
        addi $t1,$t1,1
shift_loop:
        srl $t0,$t0,1
        bnez $t0,check_loop

        li $v0,4
        la $a0,str2
        syscall

        li $v0,1
        move $a0,$t1
        syscall

        li $v0,10
        syscall
```

Input 9 and result is 2:

```
Please enter a integer: 9
Number of 1's in binary representation: 2
-- program is finished running --
```

Tast2:

2. Write a program that asks the user to enter two integers: n1 and n2 and prints the sum of all numbers from n1 to n2. For example, if the user enters n1=3 and n2=7, then the program should display the sum as 25.

Solution:

Code:

```
.data
    input_prompt_n1:    .asciiz "Enter the first integer (n1): "
    input_prompt_n2:    .asciiz "Enter the second integer (n2): "
    output_result:      .asciiz "Sum of numbers from n1 to n2: "

.text
    # Prompt the user for n1
    li $v0, 4
    la $a0, input_prompt_n1
    syscall

    # Read n1 from the user
    li $v0, 5
    syscall
    move $t0, $v0  # Store n1 in $t0

    # Prompt the user for n2
    li $v0, 4
    la $a0, input_prompt_n2
    syscall

    # Read n2 from the user
    li $v0, 5
    syscall
    move $t1, $v0  # Store n2 in $t1

    # Initialize the sum to 0
    li $t2, 0
```

```
    # Loop to calculate the sum
sum_loop:
    add $t2, $t2, $t0  # Add n1 to the sum
    addi $t0, $t0, 1   # Increment n1

    # Check if n1 is greater than n2
    bgt $t0, $t1, sum_done

    # Repeat the loop
    j sum_loop

sum_done:
    # Display the result
    li $v0, 4
    la $a0, output_result
    syscall

    # Display the sum (in $t2)
    li $v0, 1
    move $a0, $t2
    syscall

    # Exit the program
    li $v0, 10
    syscall
```

Test & screenshot:

Input 3 and 7, the output is 25:

```
Number of 1's in binary representation: 2
-- program is finished running --

Enter the first integer (n1): 3
Enter the second integer (n2): 7
Sum of numbers from n1 to n2: 25
-- program is finished running --
```

Tast3:

3. Write a program that asks the user to enter an integer and then display the hexadecimal representation of that integer.

Solution:

Code:

Lab04.asm | Lab04_tast2.asm | Lab04_task3.asm

```
.data
    input_prompt:   .asciiz "Enter an integer: "
    output_result:  .asciiz "Hexadecimal representation: "

.text
    # Prompt the user for input
    li $v0, 4
    la $a0, input_prompt
    syscall

    # Read the integer input from the user
    li $v0, 5
    syscall
    move $t0, $v0  # Store the input in $t0

    # Display the result
    li $v0, 4
    la $a0, output_result
    syscall

    # Convert and display the hexadecimal representation
    li $v0, 34  # Syscall number for "print hex"
    move $a0, $t0
    syscall

    # Exit the program
    li $v0, 10
    syscall
```

Text & screenshot:

Input 16 and the result should is 0x00000010

Mars Messages | Run I/O

Clear

```
Enter the second integer (n2): 7
Sum of numbers from n1 to n2: 25
-- program is finished running --

Enter an integer: 16
Hexadecimal representation: 0x00000010
-- program is finished running --
```

Tast4:

4. The Fibonacci sequence are the numbers in the following integer sequence: 0, 1, 1, 2, 3, 5, 8, 13, 21, 34, 55, 89, 144, ...

   Write a program that asks the user to enter a positive integer number *n* then prints the *n*-th Fibonacci number. The following algorithm can be useful:

   Input: n positive integer

   Output: n-th Fibonacci number

Pseudocode
Fib0 = 0, Fib1 = 1
for (i=2; i<=n; i++) do
temp = fib0
fib0 = fib1
fib1 = temp + fib1
if (n > 0)
fib = fib1
else fib = 0

Solution:

Code:

Edit | Execute

Lab04.asm | Lab04_tast2.asm | Lab04_task3.asm | Lab04_tast4.asm*

```
# FibonacciAssembly program to calculate the n-th Fibonacci number

.data
    input_prompt:   .asciiz "Enter a positive integer (n): "
    output_result:  .asciiz "The n-th Fibonacci number: "

.text
    # Prompt the user for input
    li $v0, 4
    la $a0, input_prompt
    syscall

    # Read the integer input from the user
    li $v0, 5
    syscall
    move $t0, $v0  # Store the input in $t0

    # Initialize the first two Fibonacci numbers
    li $t1, 0  # F(0)
    li $t2, 1  # F(1)

    # Calculate the n-th Fibonacci number
    beqz $t0, done  # If n is 0, skip the loop

fib_loop:
    add $t3, $t1, $t2  # F(n) = F(n-1) + F(n-2)
    move $t1, $t2      # Shift values for next iteration
    move $t2, $t3
    addi $t0, $t0, -1  # Decrement n
    bnez $t0, fib_loop
```

```
done:
    # Display the result
    li $v0, 4
    la $a0, output_result
    syscall

    # Display the n-th Fibonacci number (in $t1)
    li $v0, 1
    move $a0, $t1
    syscall

    # Exit the program
    li $v0, 10
    syscall
```

Test & screenshot: