CS224

Section No.: 6

Spring 2020

Lab No. 1

Osama Tanveer / 21801147

1

```
        .text
        .globl __start

__start:

        # Prompting the user to enter the number of integers
        li $v0, 4
        la $a0, promptForN
        syscall

        # Get the number of integers
        li $v0,  5
        syscall

        # Moving the value of n to s0
        addi $s0, $v0, 0

        # t0 is used to address the array
        addi $t0, $0, 0

        # Loading base address of array into t1
        la $t1, array

        # t2 used to keep track of the value of i
        addi $t2, $0, 0

        # Prompting the user to enter the values to be stored in array
        li $v0, 4
        la $a0, promptForGettingValues
        syscall

        # For loop for taking input into the array depending on the value of n
forToGetInput:
        beq $t2, $s0, inputsTaken
        li $v0, 5
        syscall
        sw $v0, array($t0)
        addi $t0, $t0, 4
        addi $t2, $t2, 1
```

```
	j forToGetInput

inputsTaken:

	# t3 is used to address the array
	li $t3, 0

	# t4 is used to keep track of the value of i
	addi $t4, $0, 0

	# s1 is used to store sum
	addi $s1, $0, 0

	li $v0, 4
	la $a0, arrayContent
	syscall
forToDisplay:
	beq $t4, $s0, displayed

	# Accessing the contents of arrat
	li $v0, 1
	lw $a0, array($t3)
	syscall

	# Computing sum
	add $s1, $s1, $a0

	# Space between the contents of array
	li $v0, 4
	la $a0, space
	syscall
	addi $t3, $t3, 4
	addi $t4, $t4, 1
	j forToDisplay
displayed:
	# Displaying sum
	li $v0, 4
	la $a0, newLine
	syscall

	li $v0, 4
	la $a0, sum
```

```
        syscall

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

        .data
                array: .space 80
                promptForN: .asciiz "Enter the number of integers: "
                promptForGettingValues: .asciiz "Enter the values to be stored in array: "
                space: .asciiz " "
                arrayContent: .asciiz "The array contents are "
                sum: .asciiz "The sum is "
                newLine: .asciiz "\n"
```

2

```
        .text
        .globl __main

__main:

        # Prompting the user to input a, b, c, d
        li $v0, 4
        la $a0, inputPrompt
        syscall

        # Getting integers a, b, c, d
        # $a0 = a, $a1 = b, $a2 = c, $a3 = d
        li $v0,  5
        syscall
        add $a0, $v0, 0

        li $v0,  5
        syscall
        add $a1, $v0, 0

        li $v0,  5
        syscall
        add $a2, $v0, 0

        li $v0,  5
```

```
    syscall
    add $a3, $v0, 0

    # Calling function computeExpression
    jal computeExpression
    move $v1, $v0

    # Output
    addi $s0, $v0, 0
    li $v0, 4
    la $a0, outputPrompt
    syscall

    li $v0, 1
    move $a0, $v1
    syscall

    # End program
    li $v0, 10
    syscall

computeExpression:
    sub $t0, $a1, $a2 # b-c
    mult $a0, $t0 # a * (b-c)
    mflo $t1 # using lower 32 bits of the product
    div $t1, $a3 # a * (b-c) / d
    mfhi $v0 # remainder stored in hi register

    jr $ra # return statement

    .data
    inputPrompt: .asciiz "Enter a, b, c, d in sequential order: "
    outputPrompt: .asciiz "The output is "
```