



# Computer Architecture

**Name:** Wajahat Ali Khan

**Sap I'd:** 55431

**Section:** BSCS 4-1

**Instructor:** Tabassum Javed

## Lab Task #03

### Task-1

- Create machine instructions and microinstructions (if required) solve the following mathematical expression. Take the first two values from use and remaining two values as hard code. (Note: First you need to write Assembly code compatible to your machine).

$$3 + 5 - 2 + 1$$

### Code:

START:

INP

STA NUM

INP

ADD NUM

STA TEMP

LDA HARD\_CODE2

CMA

ADD HARD\_CODE1

ADD TEMP

STA TEMP

LDA HARD\_CODE1

ADD TEMP

OUT

HLT

; Data storage

NUM: .data 1 0

TEMP: .data 1 0

HARD\_CODE1: .data 1 1

HARD\_CODE2: .data 1 2

### Output:

```
EXECUTING...
Enter Inputs, the first of which must be an Integer: 3
Enter Inputs, the first of which must be an Integer: 5
Output: 7
EXECUTION HALTED NORMALLY due to the setting of the bit(s): [HALT-BIT]
```

### Task-2

- Create machine instructions that runs the following Assembly code on your machine. (You need to convert this MIPS Assembly to the Assembly compatible to your machine)

**.data**

**.text**

**.globl main**

**main:**

**# Read first number**

```

    li $v0, 5
    syscall
    move $t0, $v0 # Store first number in $t0
    # Read second number
    li $v0, 5
    syscall
    move $t1, $v0 # Store second number in $t1
    # Perform addition
    add $t2, $t0, $t1 # $t2 = $t0 + $t1
    # Print sum
    li $v0, 1 move $a0, $t2
    syscall
    # Exit
    li $v0, 10
    syscall

```

#### Code:

START:

INP

STA NUM

INP

STA TEMP

LDA NUM

ADD TEMP

STA TEMP

OUT

HLT

NUM: .data 1 0

TEMP: .data 1 0

## Output:

---

```
EXECUTING...
Enter Inputs, the first of which must be an Integer: 12
Enter Inputs, the first of which must be an Integer: 7
Output: 19
EXECUTION HALTED NORMALLY due to the setting of the bit(s): [HALT-BIT]
```