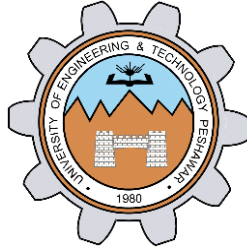


# LAB# 2



**Fall 2023**

**COA Lab**

**Submitted by: Abdul Rasheed**  
**Registration No: 21PWCSE2063**  
**Class Section: B**

“On my honor, as student of University of Engineering and Technology, I have neither given nor received unauthorized assistance on this academic work.”

Student Signature: \_\_\_\_\_

Submitted to:  
**Dr. Bilal Habib**

**Department of Computer Systems Engineering**  
**University of Engineering and Technology, Peshawar**

## BRANCHING OPERATION:

Q NO 1: Enter a number 5432 from user and then display the last digit in the console. (hint: use mfhi ).

```
.text

.globl main
main:
li $v0 ,4
la $a0, msg1
syscall

li $v0 ,5
syscall
move $t0,$v0

div $t2,$t0,10
li $v0, 4
la $a0, msg2
syscall
mfhi $t3
syscall
li $v0,1
move $a0,$t3
syscall

.data
msg1: .asciiz "Enter 1st value:"
msg2: .asciiz "Value is:"
```

Output:

```
Enter 1st value:345
Value is:Value is:5
-- program is finished running (dropped off bottom) --
```

Q NO 2: Check whether a number input by user is negative or equal to zero or greater than zero using branching ( Use bgt or ble ).

```
.text
.globl main
main:

    li $v0, 4
    la $a0, msg1
    syscall

    li $v0, 5
    syscall
    move $t0, $v0
    bgt $t0, $zero, greater
    ble $t0, $zero, less
greater:

    li $v0, 4
    la $a0, msg2
    syscall
    j end
    j end
less:
    li $v0, 4
    la $a0, msg3
    syscall

end:
    # Exit the program
    li $v0, 10
    syscall

.data
msg1: .asciiz "Enter a number: "
msg2: .asciiz "The number is greater than zero."
msg3: .asciiz "The number is less than or equal to zero."
|
```

Output:

```
Enter a number: 3
The number is greater than zero.
-- program is finished running --
```

Q NO 3: Check using branch whether the number input by user are equal or not ( Use beq ).

```
.text
.globl main
main:
    li $v0, 4
    la $a0, msg1
    syscall

    li $v0, 5
    syscall
    move $t0, $v0

    li $v0, 4
    la $a0, msg2
    syscall
    li $v0, 5
    syscall
    move $t1, $v0
    beq $t0, $t1, equal
not_equal:
```

```

not_equal:
    li $v0, 4
    la $a0, msg3
    syscall
    j end

equal:

    li $v0, 4
    la $a0, msg4
    syscall

end:|
    # Exit the program
    li $v0, 10
    syscall

.data
msg1: .asciiz "Enter the first value: "
msg2: .asciiz "Enter the second value: "
msg3: .asciiz "The values are not equal."
msg4: .asciiz "The values are equal."

```

Output:

```

Enter the first value: 3
Enter the second value: 4
The values are not equal.
-- program is finished running --

```

Q NO 4 : Write the assembly of the below C++ code.

```

Int age;
Cout<<"enter your age"<<endl;
Cin>>age;
If(age > 18)
{
Cout<<"you can apply for CNIC"<<endl;
}

```

```

Else
{
Cout<<"you cannot apply for CNIC"<<endl;
}

```

```

.text
.globl main
main:
li $v0, 4
la $a0, msg1
syscall

li $v0, 5
syscall
move $t0, $v0
li $t1, 19
bge $t0, $t1, apply

li $v0, 4
la $a0, msg3
syscall
j end

```

```

apply:
li $v0, 4
la $a0, msg2
syscall

end:
li $v0, 10
syscall

.data
msg1: .asciiz "Enter your age: "
msg2: .asciiz "You can apply for CNIC."
msg3: .asciiz "You cannot apply for CNIC."

```

Output:

```

Enter your age: 3
You cannot apply for CNIC.
-- program is finished running --

```

Q NO 5: Write a program which take a limit from user and compute the sum of numbers from 0 to the limit ( Use bqe, add, addi, and J (jump)).

Below is the C++ language code:

```
Int limit; Int
```

```
sum;
```

```
Cout<<"Enter a number"<<endl; Cin>>limit;
```

```
for (int i = 1; i <= limit; ++i)
{    sum += i;    }
```

```
Cout<<"sum of numbers from 1 to <<limit<<"is"<<sum<<endl;
```

```
|.text
.globl main
```

```
main:
```

```
li $v0, 4
la $a0, msg1
syscall
```

```
addi $v0, $zero, 5
syscall
move $t0, $v0
```

```
addi $t1, $zero, 0
loop:
bge $t1, $t0, done
addi $t1, $t1, 1
lw $t2, sum
add $t2, $t2, $t1
sw $t2, sum
j loop
```

```
done:
```

```
addi $v0, $zero, 1
lw $a0, sum
syscall
```

```
# Exit the program.
addi $v0, $zero, 10
syscall
```

```
.data
```

```
msg1: .asciiz "Enter a number: "
sum: .word 0
```

Output:

```
Enter a number: 3
6
-- program is finished running --
```