Computer Organization And Architecture

Lab-4 -Loops

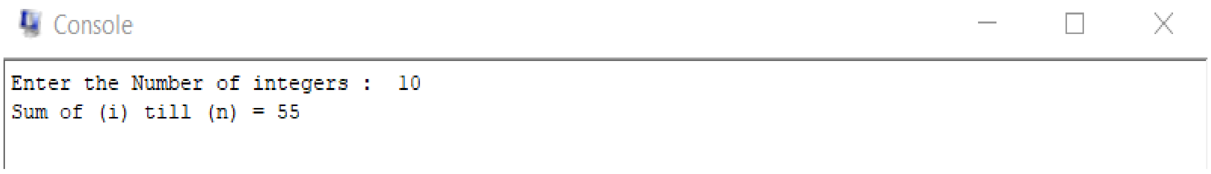
S Abhishek

AM.EN.U4CSE19147

1. Sample code for a loop to compute the sum of N integers: 1 + 2 + ... + N

```
.data
	input: .asciiz "Enter the Number of integers :  "
	output: .asciiz "Sum of (i) till (n) = "

.text
	.globl main
	main:

		li $v0,4
		la $a0,input
		syscall

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

```
          li $t1,0
          li $t2,0

Continue:

          addi $t1,$t1,1
          add $t2,$t2,$t1
          beq $t0,$t1,Quit
          j Continue

Quit:
          li $v0,4
          la $a0,output
          syscall

          li $v0,1
          move $a0,$t2
          syscall

          li $v0,10
          syscall
```

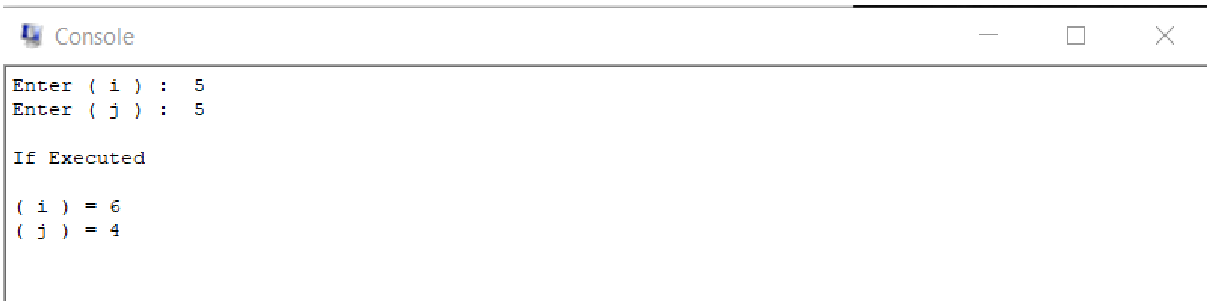


```
Enter the Number of integers :  10
Sum of (i) till (n) = 55
```

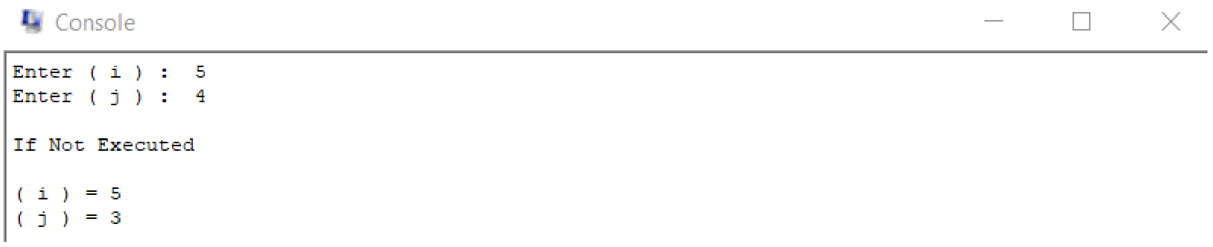
2. Convert the following c-like code into MIPS assembly code.

```
if ( i == j )
    i++ ;
j-- ;
```

```
.data
        input1: .asciiz "Enter ( i ) :  "
        input2: .asciiz "Enter ( j ) :  "
        output1: .asciiz "\n( i ) = "
        output2: .asciiz "\n( j ) = "

        prompt1: .asciiz "\nIf Executed\n"
        prompt2: .asciiz "\nIf Not Executed\n"

.text

        .globl main

        main:

                li $v0,4
                la $a0,input1
                syscall
        li $v0,5
        syscall
        move $t0,$v0 #i

        li $v0,4
        la $a0,input2
        syscall

        li $v0,5
        syscall
        move $t1,$v0 #j

        beq $t0,$t1,Increment

        li $v0,4
        la $a0,prompt2
        syscall

        j Exit

Increment:

        li $v0,4
        la $a0,prompt1
        syscall
        addi $t0,$t0,1
        j Exit

Exit:

        addi $t1,$t1,-1

        li $v0,4
        la $a0,output1 #Sample Checking of ( i )
        syscall

        li $v0,1
        move $a0,$t0
        syscall

        li $v0,4
        la $a0,output2 #Sample Checking of ( j )
        syscall

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

        li $v0,10
        syscall
```



```
Enter ( i ) :  5
Enter ( j ) :  5

If Executed

( i ) = 6
( j ) = 4
```



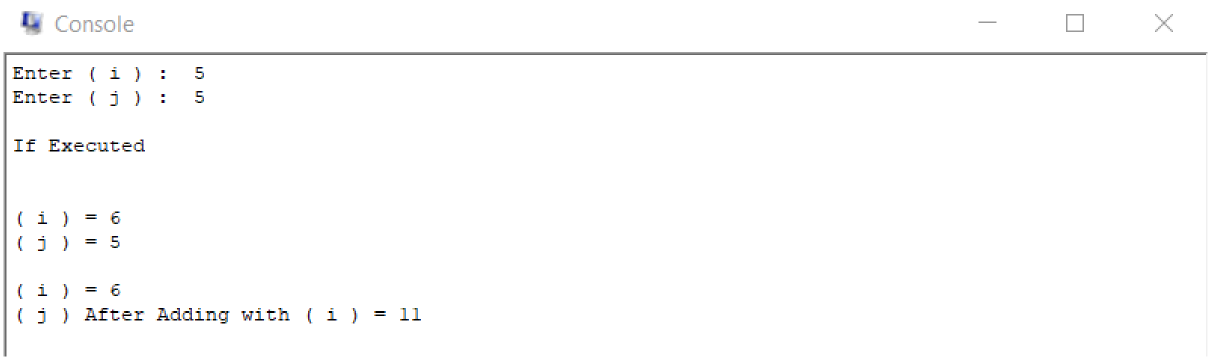
```
Enter ( i ) :  5
Enter ( j ) :  4

If Not Executed

( i ) = 5
( j ) = 3
```

3.Convert the following c-like code into MIPS assembly code.

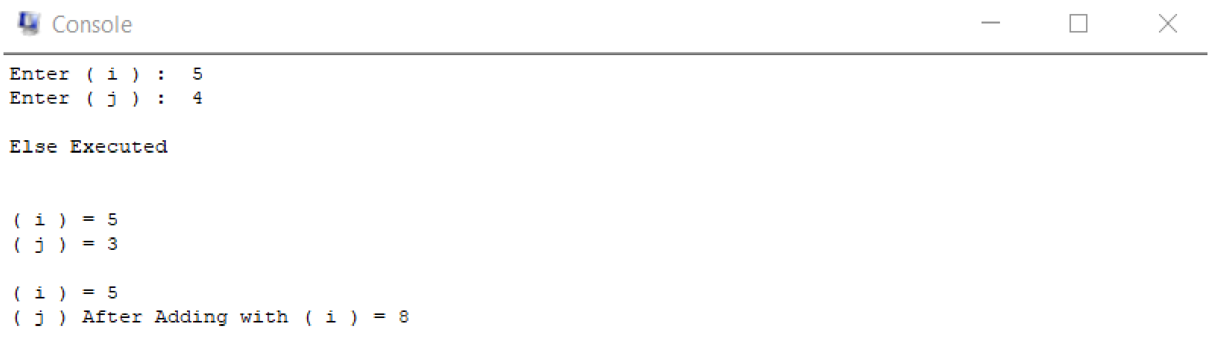
```
if ( i == j )
        i++ ;
else
      j-- ;
j += i ;
```

```
.data
        input1: .asciiz "Enter ( i ) :  "
        input2: .asciiz "Enter ( j ) :  "
        output1: .asciiz "\n\n( i ) = "
        output2: .asciiz "\n( j ) = "
        output3: .asciiz "\n( j ) After Adding with ( i ) = "
        prompt1: .asciiz "\nIf Executed\n"

        prompt2: .asciiz "\nElse Executed\n"


.text


        .globl main


        main:


                li $v0,4

                la $a0,input1

                syscall


                li $v0,5

                syscall

                move $t0,$v0 #i


                li $v0,4

                la $a0,input2

                syscall


                li $v0,5

                syscall

                move $t1,$v0 #j
        beq $t0,$t1,Increment

        bne $t0,$t1,Decrement


Increment:


        addi $t0,$t0,1


        li $v0,4

        la $a0,prompt1

        syscall


        li $v0,4 #Sample Checking of ( i )

        la $a0,output1

        syscall


        li $v0,1

        move $a0,$t0

        syscall


        li $v0,4 #Sample Checking of ( j )

        la $a0,output2

        syscall


        li $v0,1

        move $a0,$t1
        syscall
        j Exit

Decrement:
        addi $t1,$t1,-1
        li $v0,4
        la $a0,prompt2
        syscall

        li $v0,4 #Sample Checking of ( i )
        la $a0,output1
        syscall

        li $v0,1
        move $a0,$t0
        syscall

        li $v0,4 #Sample Checking of ( j )
        la $a0,output2
        syscall

        li $v0,1
        move $a0,$t1
        syscall
        j Exit
```

```
Exit:

        add $t1,$t1,$t0

        li $v0,4 #Sample Checking of ( i )
        la $a0,output1
        syscall

        li $v0,1
        move $a0,$t0
        syscall

        li $v0,4 #Sample Checking of ( j )
        la $a0,output3
        syscall

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

        li $v0,10
        syscall
```



```
Enter ( i ) :  5
Enter ( j ) :  5

If Executed


( i ) = 6
( j ) = 5

( i ) = 6
( j ) After Adding with ( i ) = 11
```



```
Enter ( i ) :  5
Enter ( j ) :  4

Else Executed


( i ) = 5
( j ) = 3

( i ) = 5
( j ) After Adding with ( i ) = 8
```

4. Convert the following c-like code into MIPS assembly code.

```
if ( i == j && i == k )
        j++ ;
        i-- ;
else
        j = i + k-2 ;
```

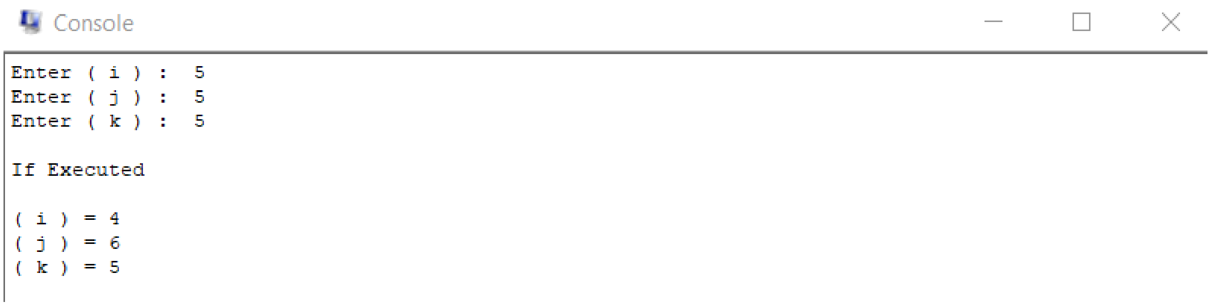
```
.data
        input1: .asciiz "Enter ( i ) :  "
        input2: .asciiz "Enter ( j ) :  "
        input3: .asciiz "Enter ( k ) :  "
        output1: .asciiz "\n( i ) = "
```

```
          output2: .asciiz "\n( j ) = "

          output3: .asciiz "\n( k ) = "


          prompt1: .asciiz "\nIf Executed\n"

          prompt2: .asciiz "\nElse Executed\n"



.text


          .globl main


          main:


                    li $v0,4

                    la $a0,input1

                    syscall


                    li $v0,5

                    syscall

                    move $t0,$v0 #i


                    li $v0,4

                    la $a0,input2

                    syscall
```

```
li $v0,5
syscall
move $t1,$v0 #j

li $v0,4
la $a0,input3
syscall

li $v0,5
syscall
move $t2,$v0 #j

bne $t0,$t1,Else
bne $t0,$t2,Else

li $v0,4
la $a0,prompt1
syscall

addi $t1,$t1,1
addi $t0,$t0,-1
j Exit
```

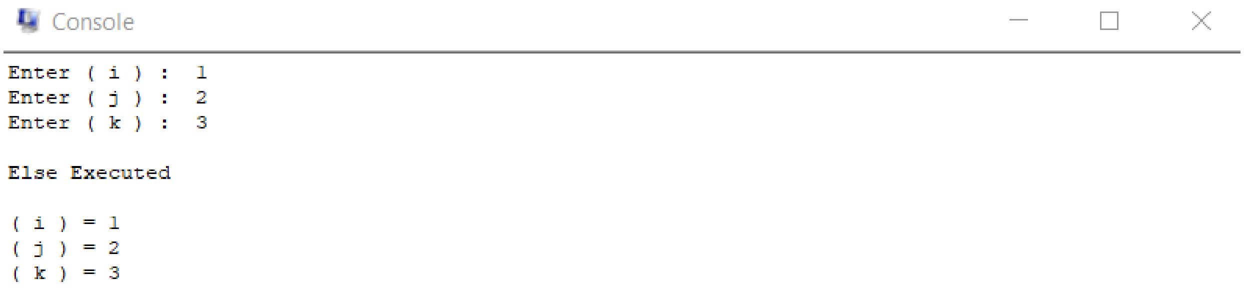
```
Else:


        li $v0,4
        la $a0,prompt2
        syscall

        addi $t1,$t0,0
        addi $t3,$t2,-2
        add $t1,$t1,$t3

Exit:
        li $v0,4
        la $a0,output1
        syscall

        li $v0,1
        move $a0,$t0
        syscall

        li $v0,4
        la $a0,output2
        syscall

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

```
syscall

li $v0,4
la $a0,output3
syscall

li $v0,1
move $a0,$t2
syscall

li $v0,10
syscall
```



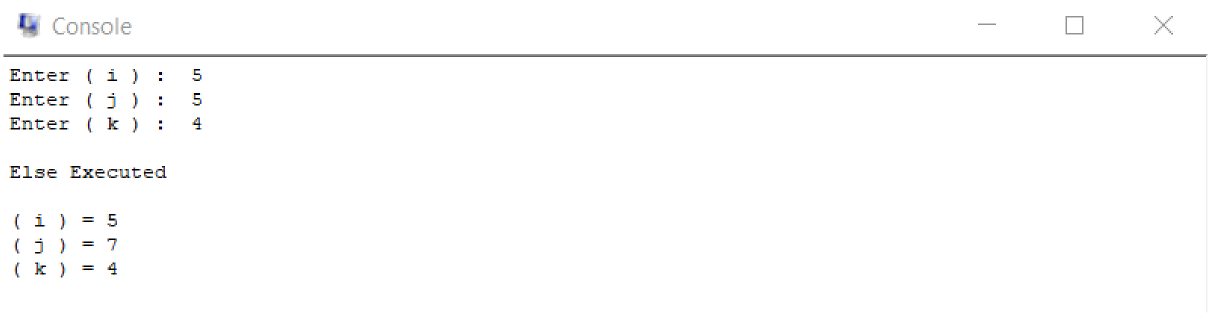
Console

```
Enter ( i ) :  5
Enter ( j ) :  5
Enter ( k ) :  5

If Executed

( i ) = 4
( j ) = 6
( k ) = 5
```



Console

```
Enter ( i ) :  1
Enter ( j ) :  2
Enter ( k ) :  3

Else Executed

( i ) = 1
( j ) = 2
( k ) = 3
```



```
Enter ( i ) :  5
Enter ( j ) :  5
Enter ( k ) :  4

Else Executed

( i ) = 5
( j ) = 7
( k ) = 4
```

5. Convert the following c-like code into MIPS assembly code.

```
if ( i==j || i==k )
        i++ ;
        j-- ;
else
        j = i + k ;
```

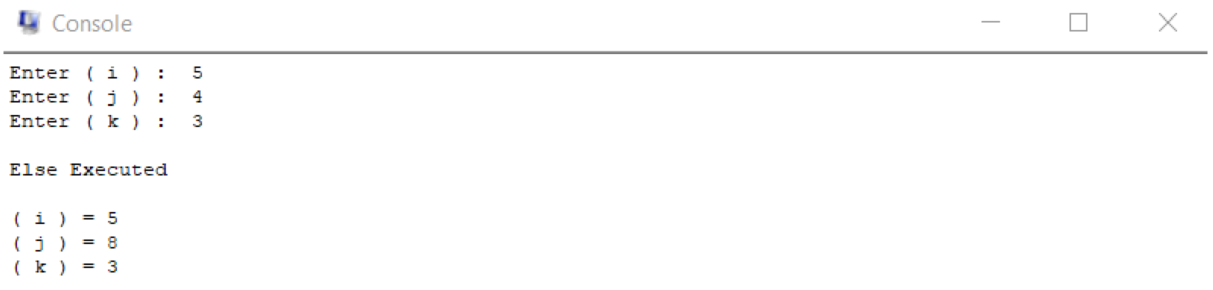
```
.data
        input1: .asciiz "Enter ( i ) :  "
        input2: .asciiz "Enter ( j ) :  "
        input3: .asciiz "Enter ( k ) :  "
        output1: .asciiz "\n( i ) = "
        output2: .asciiz "\n( j ) = "
        output3: .asciiz "\n( k ) = "

        prompt1: .asciiz "\nIf Executed\n"
        prompt2: .asciiz "\nElse Executed\n"
```

```
.text

        .globl main

        main:

                li $v0,4
                la $a0,input1
                syscall

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

                li $v0,4
                la $a0,input2
                syscall

                li $v0,5
                syscall
                move $t1,$v0 #j

                li $v0,4
                la $a0,input3
                syscall
```

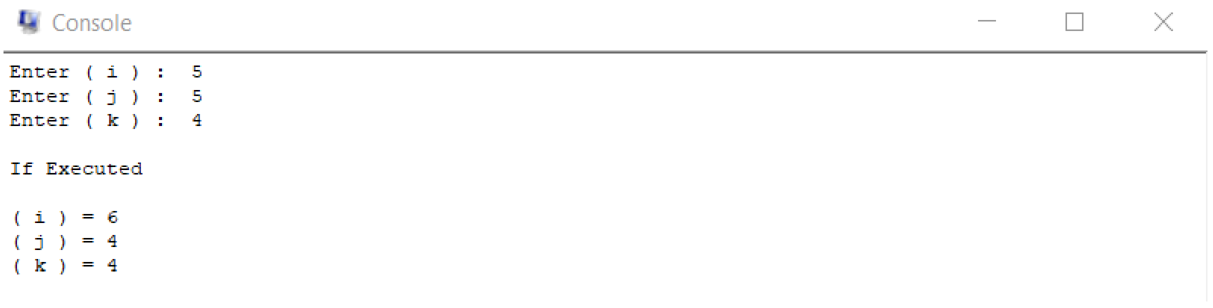
```
          li $v0,5
          syscall
          move $t2,$v0 #j

          beq $t0,$t1,If
          beq $t0,$t2,If

          li $v0,4
          la $a0,prompt2
          syscall

          add $t1,$t0,$t2

          j Exit


      If:

          li $v0,4
          la $a0,prompt1
          syscall

          addi $t0,$t0,1
          addi $t1,$t1,-1
```

```
Exit:

        li $v0,4
        la $a0,output1
        syscall

        li $v0,1
        move $a0,$t0
        syscall

        li $v0,4
        la $a0,output2
        syscall

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

        li $v0,4
        la $a0,output3
        syscall

        li $v0,1
        move $a0,$t2
        syscall
```

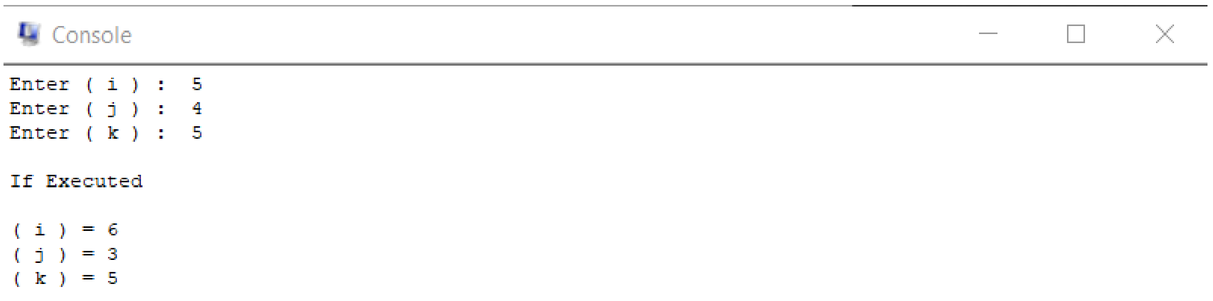
```
li $v0,10

syscall
```



```
Enter ( i ) :  5
Enter ( j ) :  4
Enter ( k ) :  3

Else Executed

( i ) = 5
( j ) = 8
( k ) = 3
```



```
Enter ( i ) :  5
Enter ( j ) :  5
Enter ( k ) :  4

If Executed

( i ) = 6
( j ) = 4
( k ) = 4
```



```
Enter ( i ) :  5
Enter ( j ) :  4
Enter ( k ) :  5

If Executed

( i ) = 6
( j ) = 3
( k ) = 5
```

Thankyou !!