Computer Organization & Architecture Lab

Lab Report # 02

Submitted By: AWAIS SADDIQUI

Registration No: 21PWCSE1993

Section: “A”

“On my honor, as student at 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.

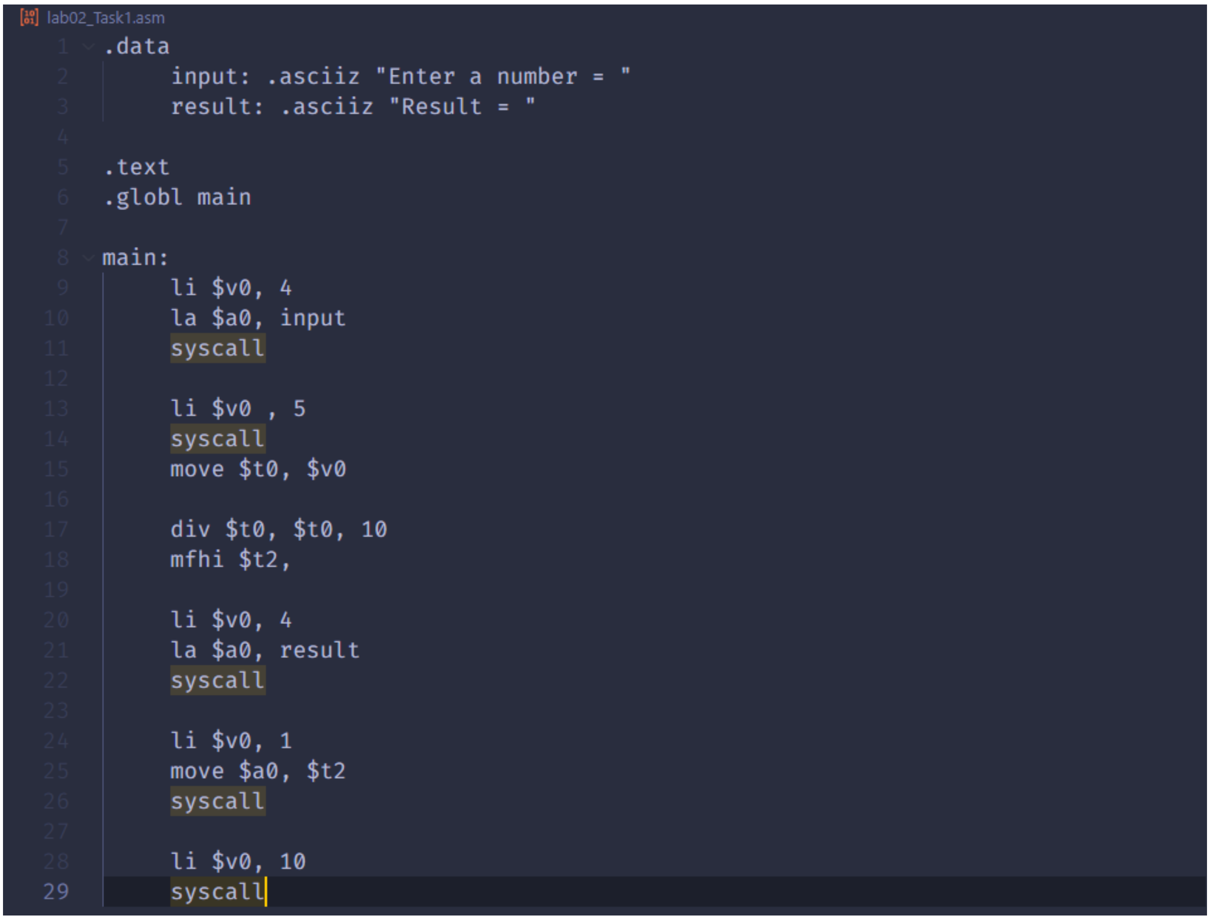
ASSESSMENT RUBRICS COA LABS

LAB REPORT ASSESSMENT				
Criteria	**Excellent**	**Average**	**Nill**	**Marks Obtained**
1. Objectives of Lab	All objectives of lab are properly covered [Marks 10]	Objectives of lab are partially covered [Marks 5]	Objectives of lab are not shown [Marks 0]	
2. MIPS instructions with Comments and proper indentations.	All the instructions are well written with comments explaining the code and properly indented	Some instructions are missing are poorly commented code [Marks 10]	The instructions are not properly written [Marks 0]	
3. Simulation run without error and warnings	The code is running in the simulator without any error and warnings [Marks 10]	The code is running but with some warnings or errors. [Marks 5]	The code is written but not running due to errors [Marks 0]	
4. Procedure	All the instructions are written with proper procedure	Some steps are missing [Marks 10]	steps are totally missing [Marks 0]	
5. OUTPUT	Proper output of the code written in assembly [Marks 20]	Some of the outputs are missing [Marks 10]	No or wrong output [Marks 0]	
6. Conclusion	Conclusion about the lab is shown and written [Marks 20]	Conclusion about the lab is partially shown	Conclusion about the lab is not shown[Marks0]	
7. Cheating			Any kind of cheating will lead to 0 Marks	
Total Marks Obtained:__________ Instructor Signature: ______________________				

BRANCHING OPERATION:

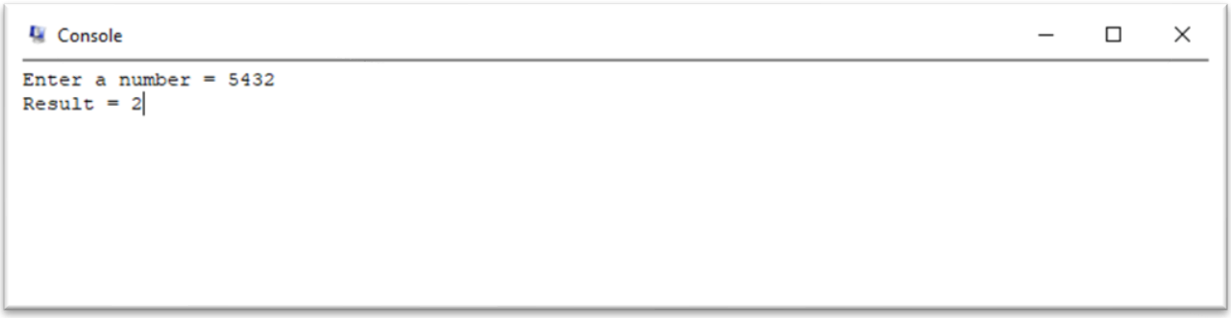
Question # 01:

Enter a number 5432 from user and then display the last digit in the console.



```
.data
     input: .asciiz "Enter a number = "
     result: .asciiz "Result = "

.text
.globl main

main:
     li $v0, 4
     la $a0, input
     syscall

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

     div $t0, $t0, 10
     mfhi $t2,

     li $v0, 4
     la $a0, result
     syscall

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

     li $v0, 10
     syscall
```

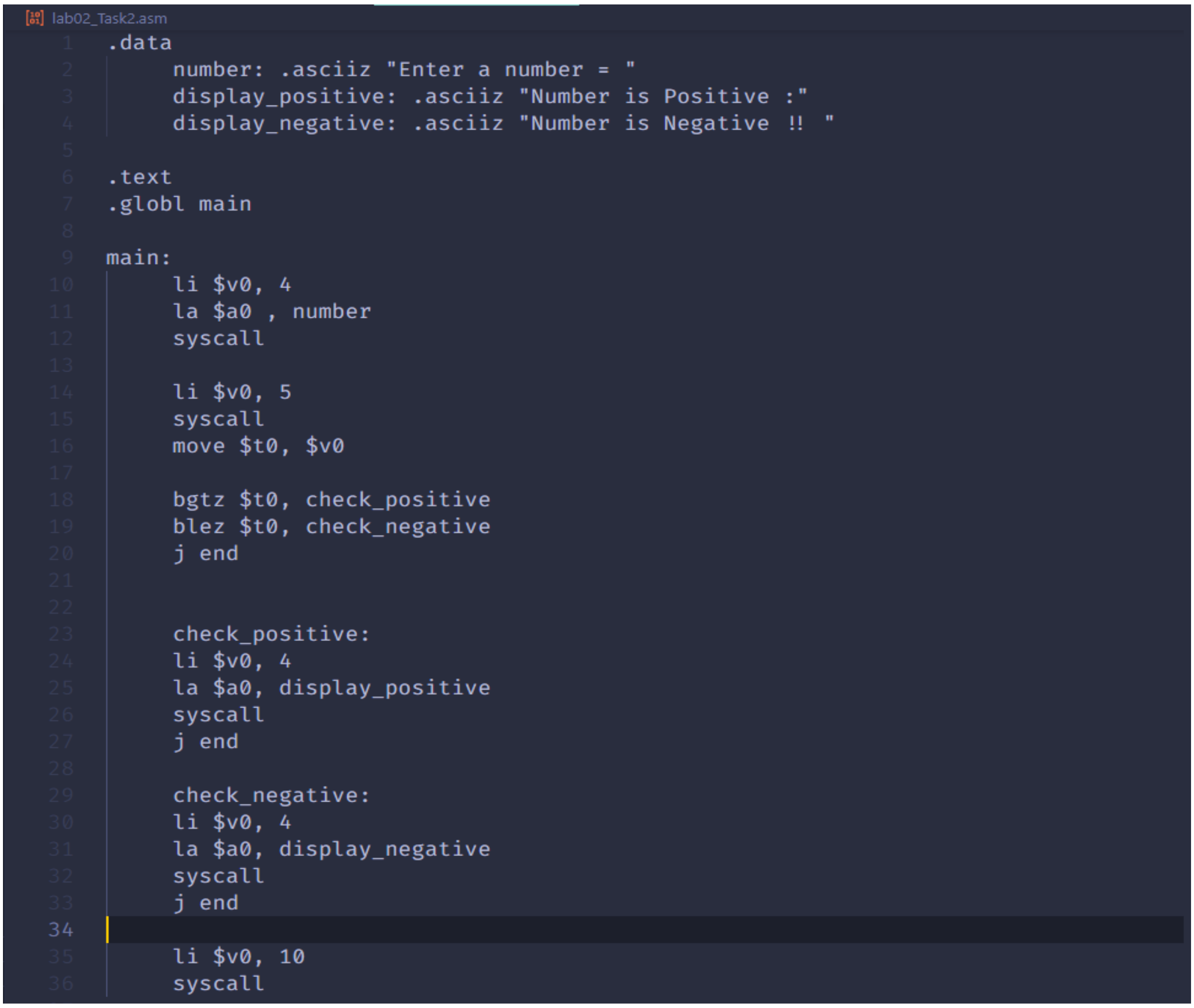
Output:



```
Enter a number = 5432
Result = 2
```

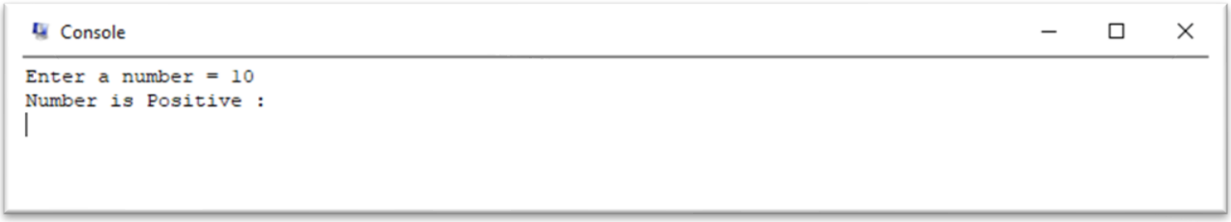
Question # 02:

Check whether a number input by user is negative or equal to zero or greater than zero using branching.

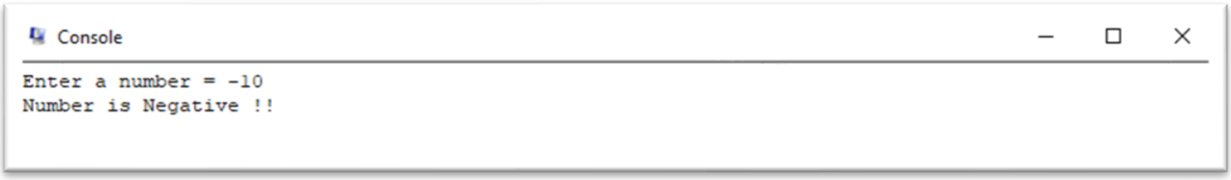


```
.data
     number: .asciiz "Enter a number = "
     display_positive: .asciiz "Number is Positive :"
     display_negative: .asciiz "Number is Negative !!  "

.text
.globl main

main:
     li $v0, 4
     la $a0 , number
     syscall

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

     bgtz $t0, check_positive
     blez $t0, check_negative
     j end


     check_positive:
     li $v0, 4
     la $a0, display_positive
     syscall
     j end

     check_negative:
     li $v0, 4
     la $a0, display_negative
     syscall
     j end

     li $v0, 10
     syscall
```

Output:



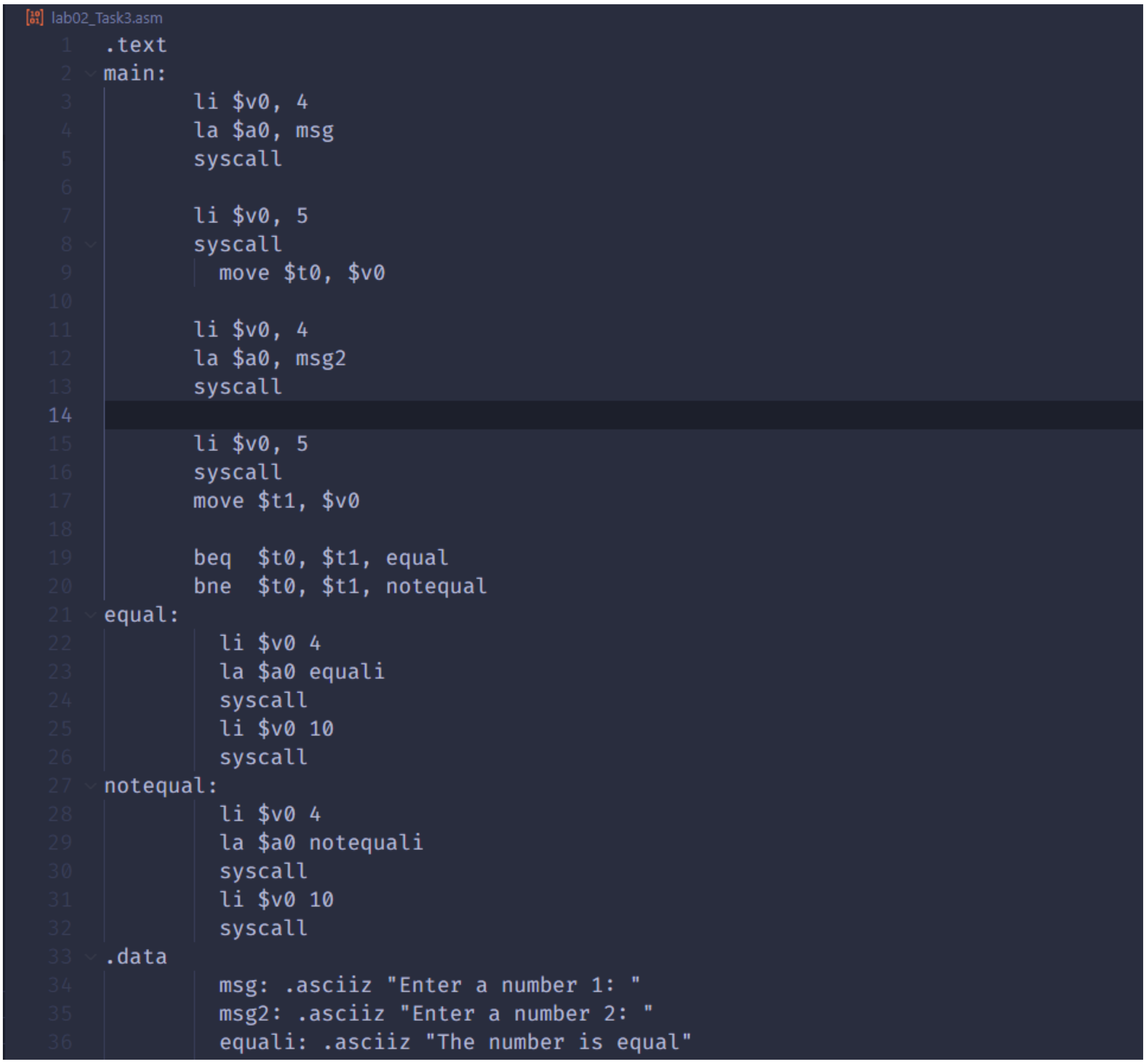
```
Enter a number = 10
Number is Positive :
```



```
Enter a number = -10
Number is Negative !!
```

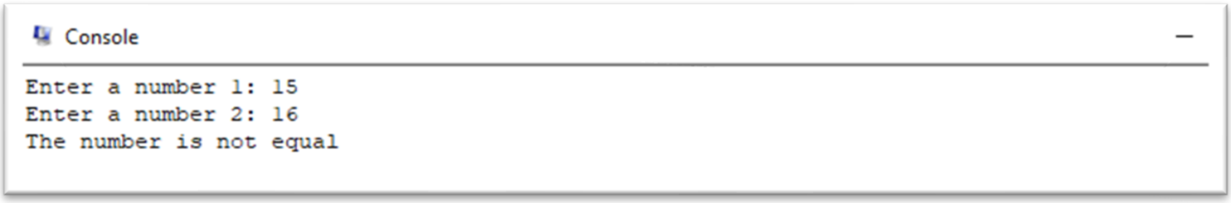
Question # 03:

Check using branch whether the number input by user is equal or not.



```
.text
main:
       li $v0, 4
       la $a0, msg
       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
       bne  $t0, $t1, notequal
equal:
         li $v0 4
         la $a0 equali
         syscall
         li $v0 10
         syscall
notequal:
         li $v0 4
         la $a0 notequali
         syscall
         li $v0 10
         syscall
.data
         msg: .asciiz "Enter a number 1: "
         msg2: .asciiz "Enter a number 2: "
         equali: .asciiz "The number is equal"
```

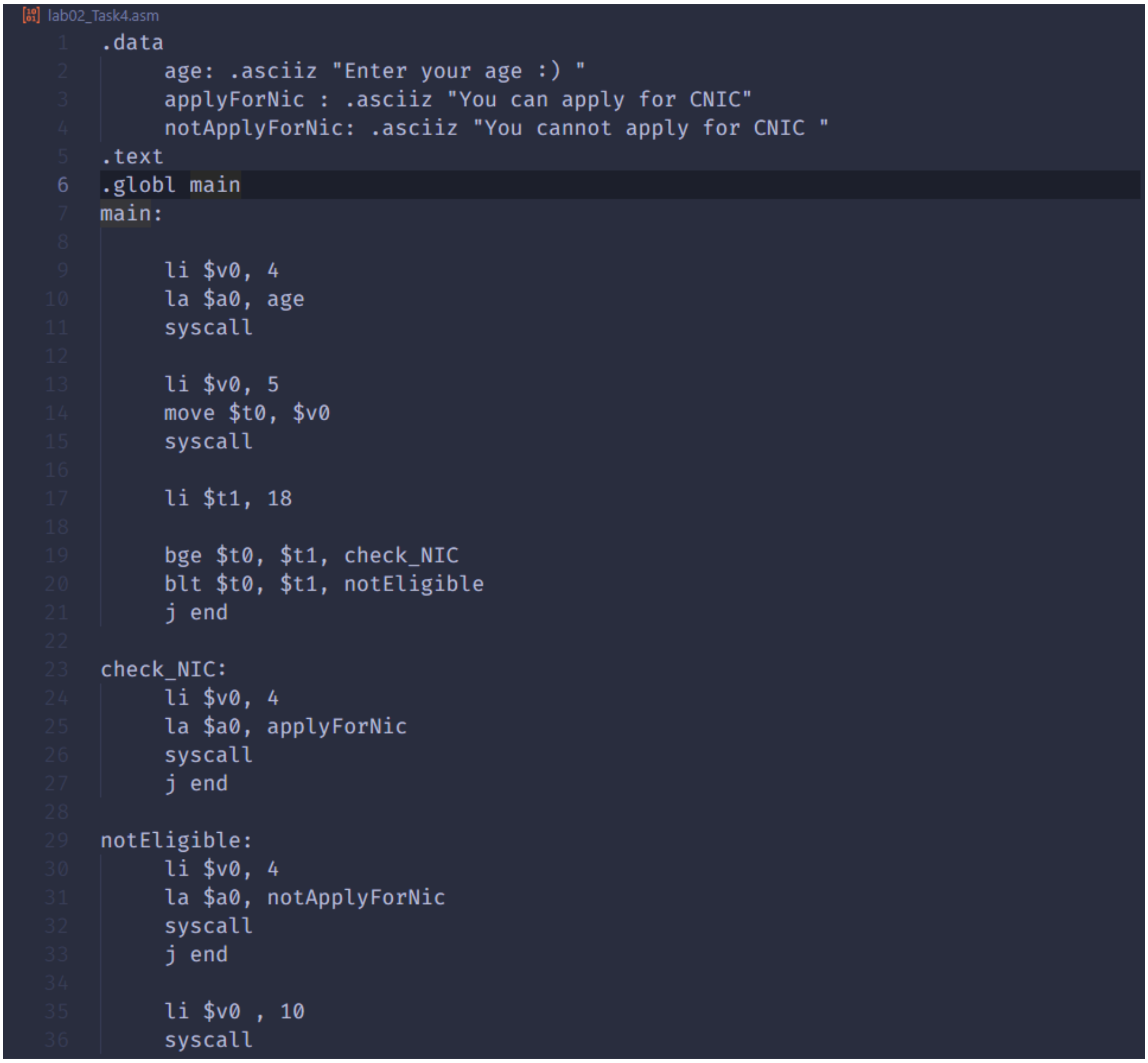
Output:



```
Enter a number 1: 15
Enter a number 2: 16
The number is not equal
```

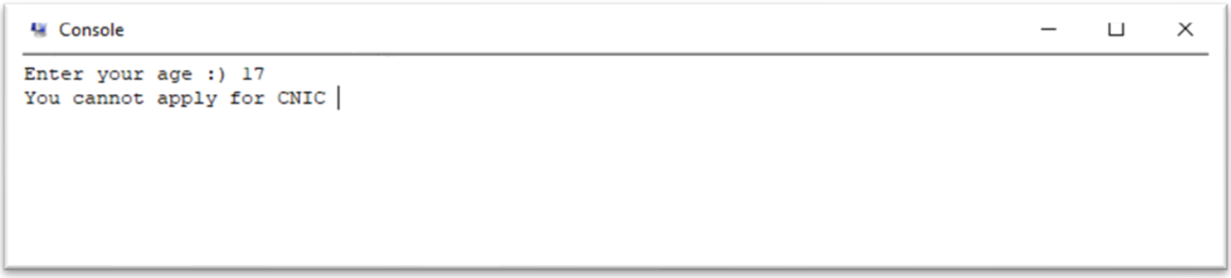
Question # 04:

Write the assembly of the below C++ code.



```
.data
	age: .asciiz "Enter your age :) "
	applyForNic : .asciiz "You can apply for CNIC"
	notApplyForNic: .asciiz "You cannot apply for CNIC "
.text
.globl main
main:

	li $v0, 4
	la $a0, age
	syscall

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

	li $t1, 18

	bge $t0, $t1, check_NIC
	blt $t0, $t1, notEligible
	j end

check_NIC:
	li $v0, 4
	la $a0, applyForNic
	syscall
	j end

notEligible:
	li $v0, 4
	la $a0, notApplyForNic
	syscall
	j end

	li $v0 , 10
	syscall
```

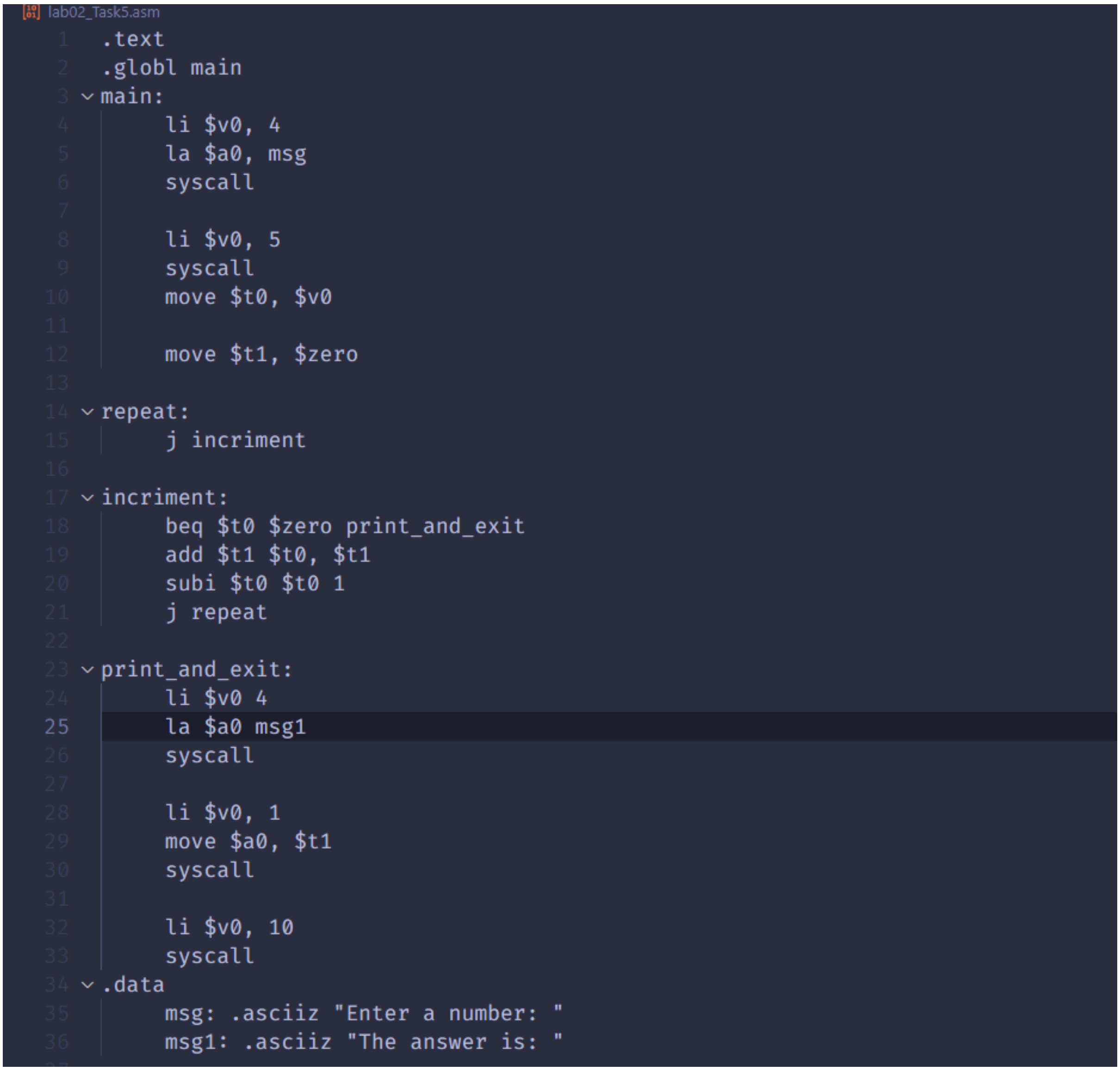
Output:



```
Enter your age :) 17
You cannot apply for CNIC
```

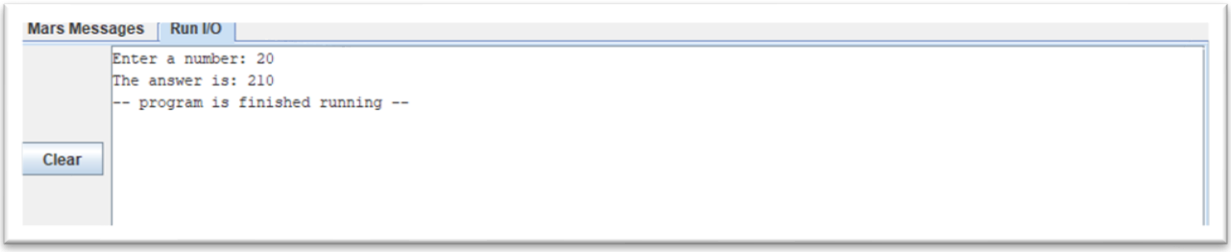
Question # 05:

Write a program which takes a limit from user and compute the sum of numbers from 0 to the limit.



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

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

     move $t1, $zero

repeat:
     j incriment

incriment:
     beq $t0 $zero print_and_exit
     add $t1 $t0, $t1
     subi $t0 $t0 1
     j repeat

print_and_exit:
     li $v0 4
     la $a0 msg1
     syscall

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

     li $v0, 10
     syscall
.data
     msg: .asciiz "Enter a number: "
     msg1: .asciiz "The answer is: "
```

Output:



```
Enter a number: 20
The answer is: 210
-- program is finished running --
```

Note: ***This code (Task_5) does not run on QTspim that’s why I use Mars.***