CS211: LAB Report

Assembly Related Labs

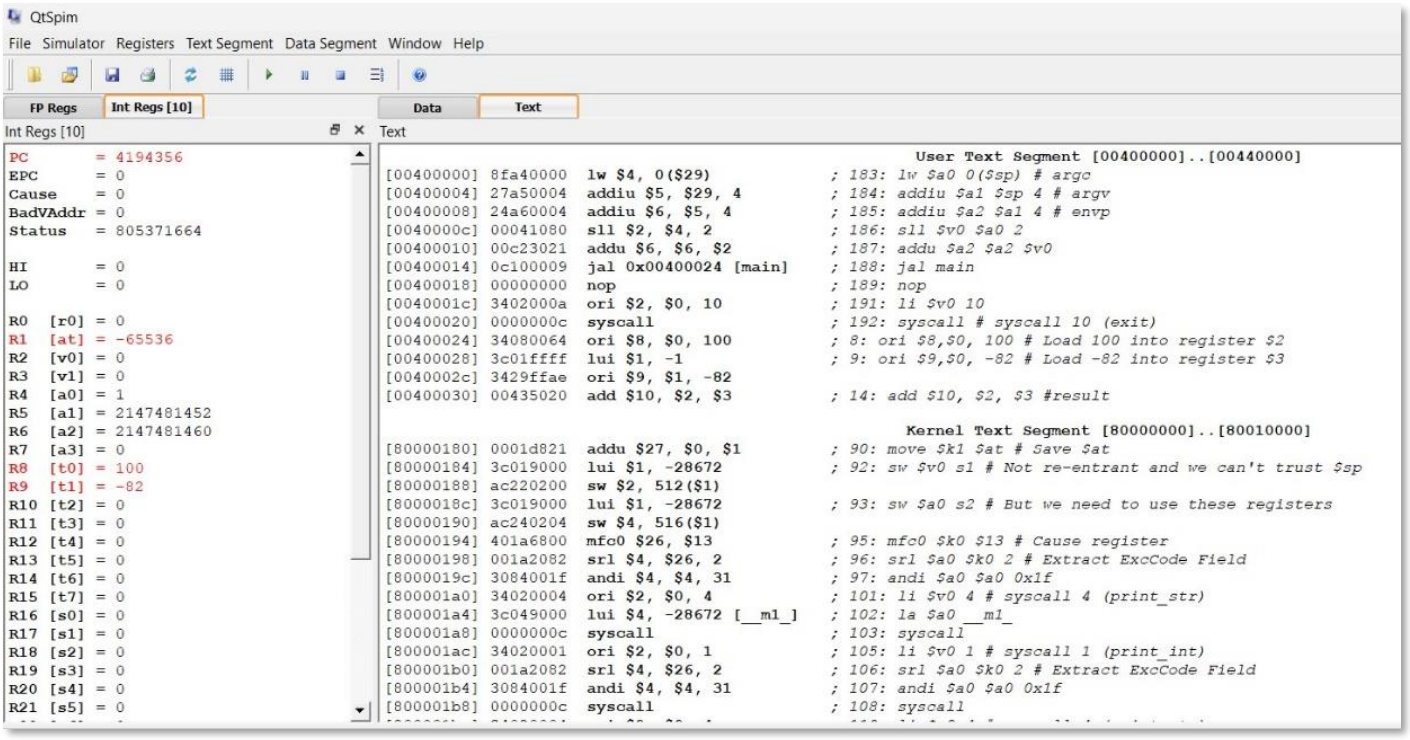
By Rupesh Bhusare, 2203106

LAB 1:

Question 1:

Add numbers 100 and -82 and store the result in register $10. (Hint: Find 2's complement of 82 and add it to 100)

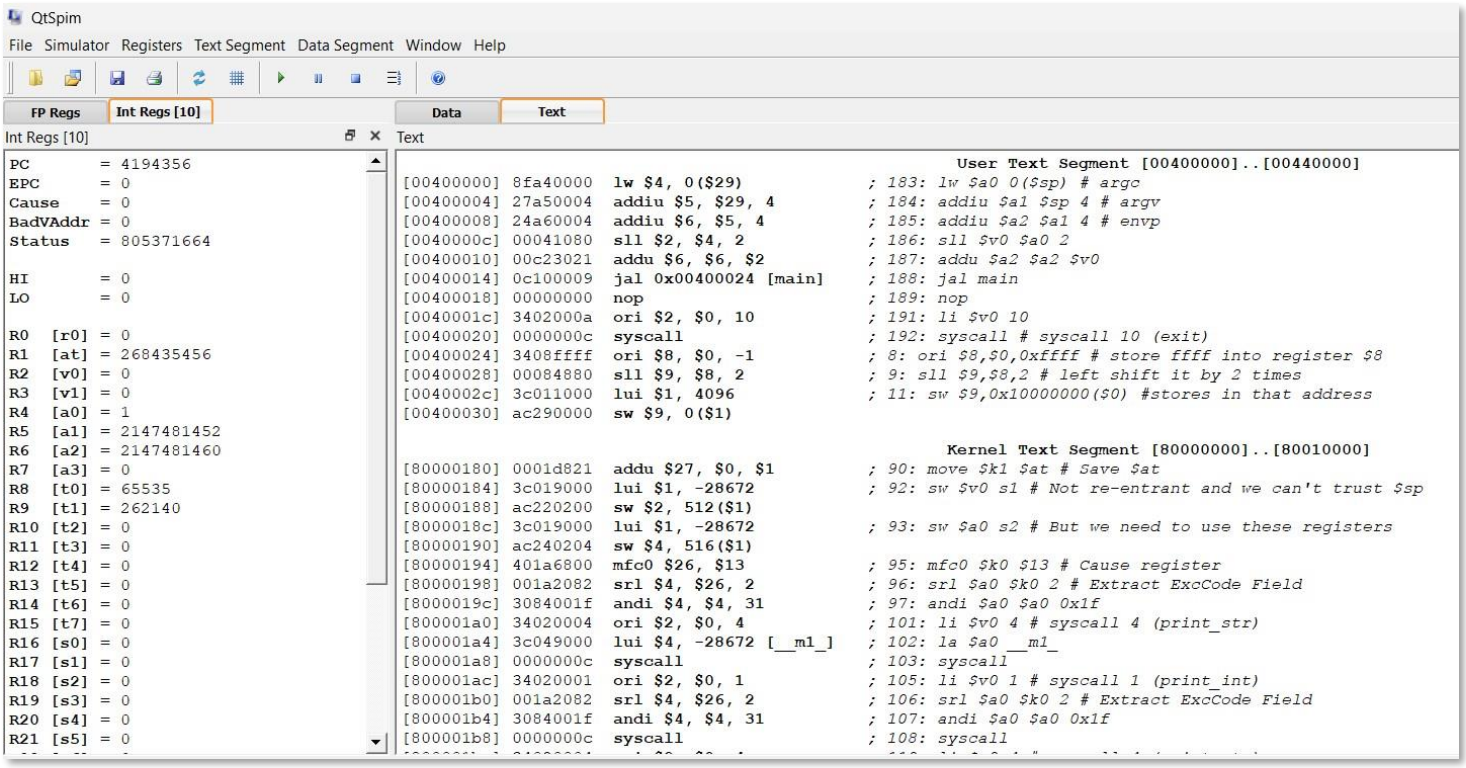
Output screenshot:



Question 2:

Store FFFF in $8. Left shift it 2 times and store the result in memory location 0x10000000.

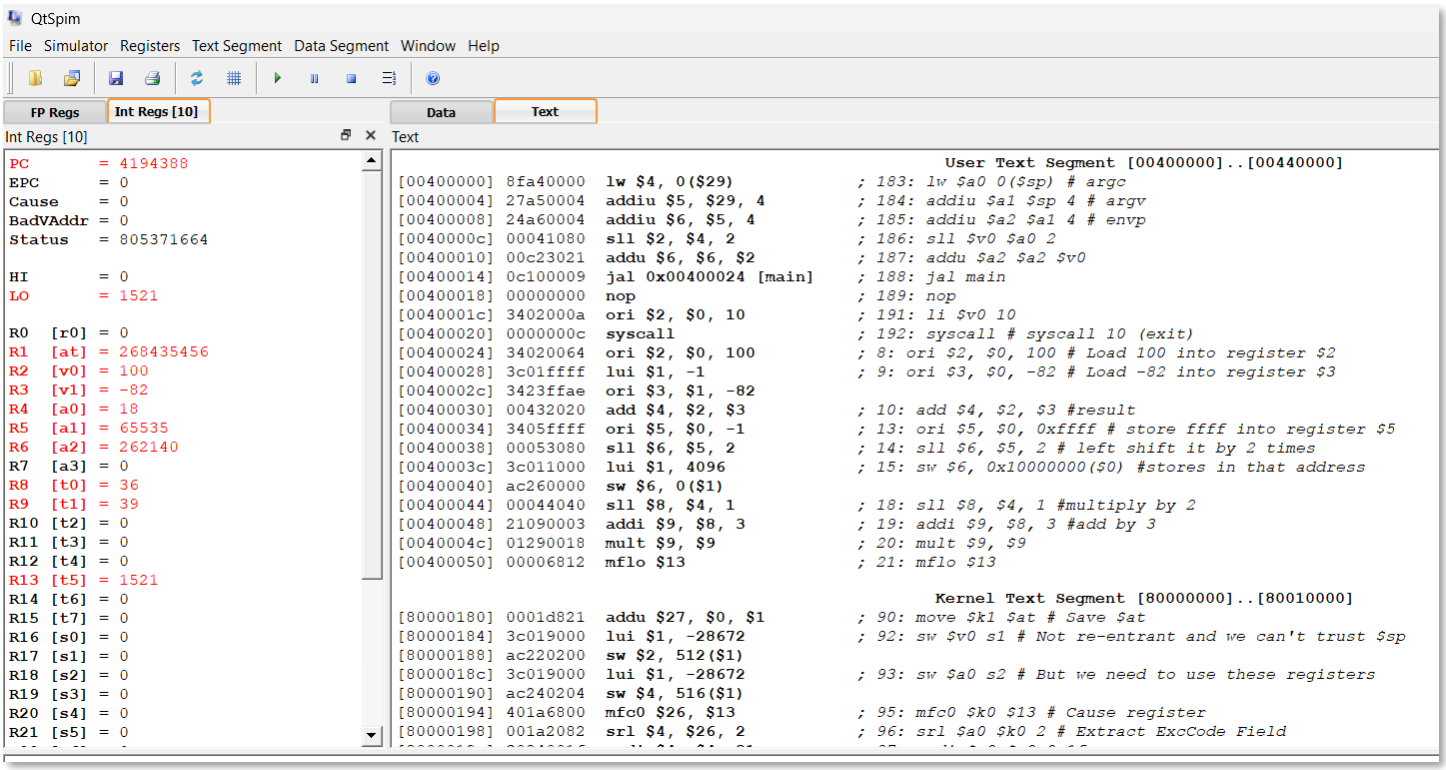
Output screenshot:



Question 3:

Evaluate the expression (2x+3) 2 where x is the content in register $10 based on exercise (a). Store the result in register $13.

Output screenshot:



LAB 2:

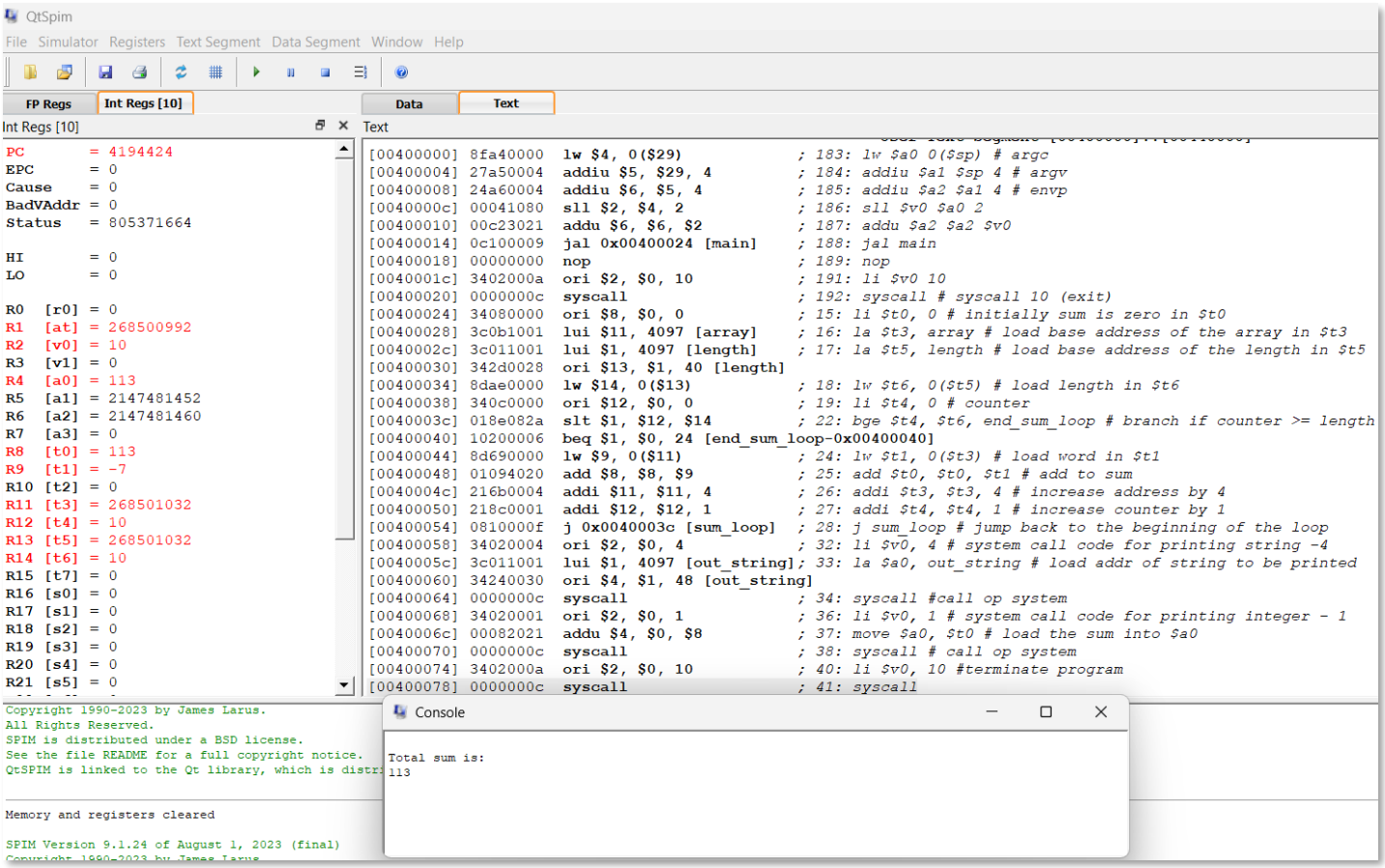
Question 1:

1) Complete the following code snippet to add 10 numbers stored consecutively in data memory. Print the result.

```
.data
array : .word  10,12,15,-10,13,82,-9,4,3,-7        #array={10,12,15,-10,13,82,-9,4,3,-7}
length:  .word 10                                  #load the length of the array as 10
sum: .word 0                                       #initialise sum to 0
.text
main:
la $t3, array                          # load base address of the array
# $t3 has the base address of data. All the subsequent data can be accessed using respective offset values.
#Add your code here
```

Count the total number of machine instructions executed to complete this task.

Output screenshot:



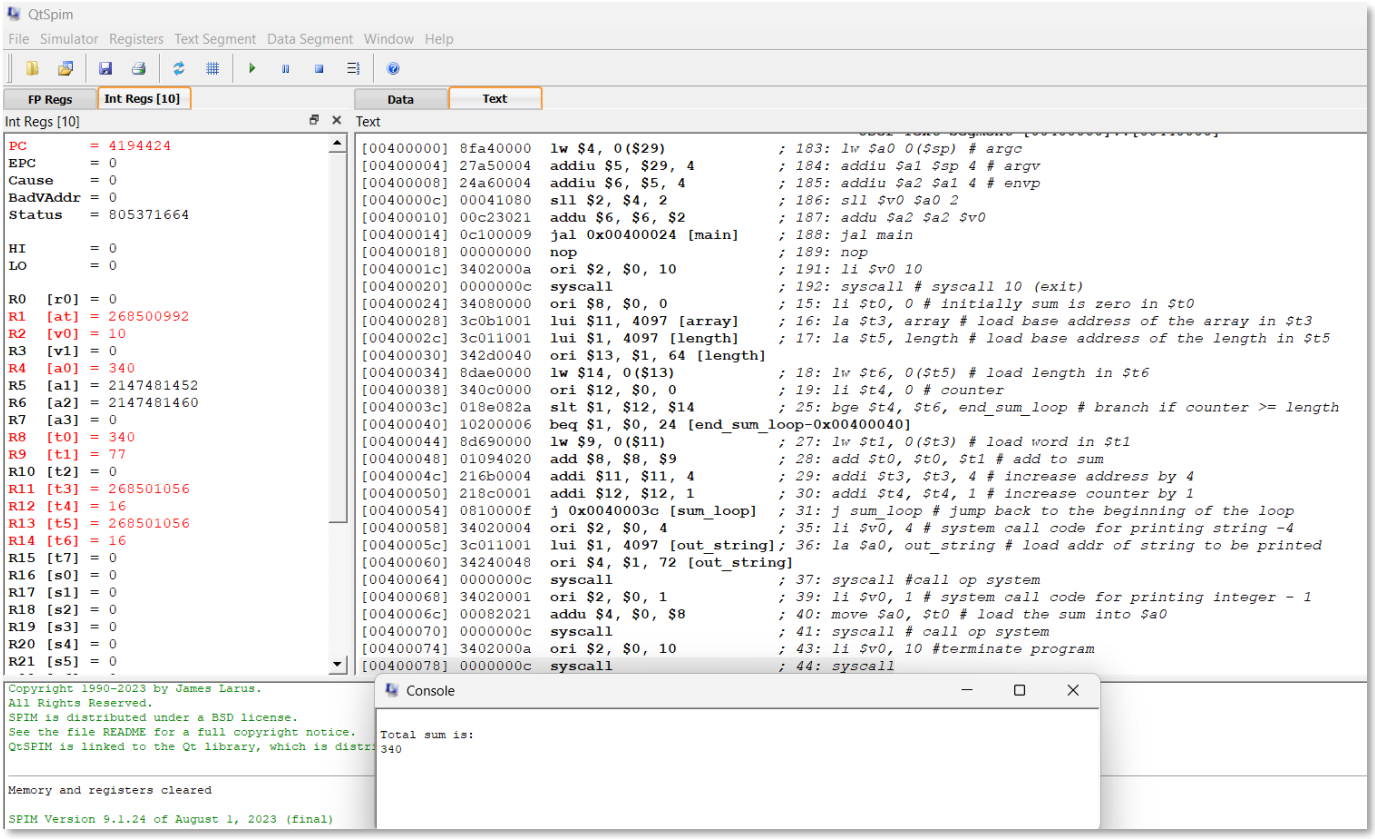
Ans: Count of total number of instructions: 59

Question 2:

2) Include the following numbers in the array data segment of question 1.
10,20,30,40,50,77

Now you have 16 numbers residing in the array (data memory). Add these numbers and display the result.
Count the total number of instructions.
Compare and analyse the relation between the number of data elements and total number of machine instructions executed.

Output screenshot:



Ans: Count of the total number of instructions: 89.

The relationship between the number of data elements and the total number of machine instructions executed is linear. The number of instructions inside the loop is executed once for each data element. So, if you increase the number of data elements, the total number of instructions executed will also increase proportionally.

Question 3:

3) Compute the Euler Phi function for the number 21.

Euler's Phi function for an input n, denoted as (phi(n)) is the count of numbers in {1, 2, 3, …, n} that are relatively prime to n, i.e, the numbers whose GCD (Greatest Common Divisor) with n is 1.
Examples:

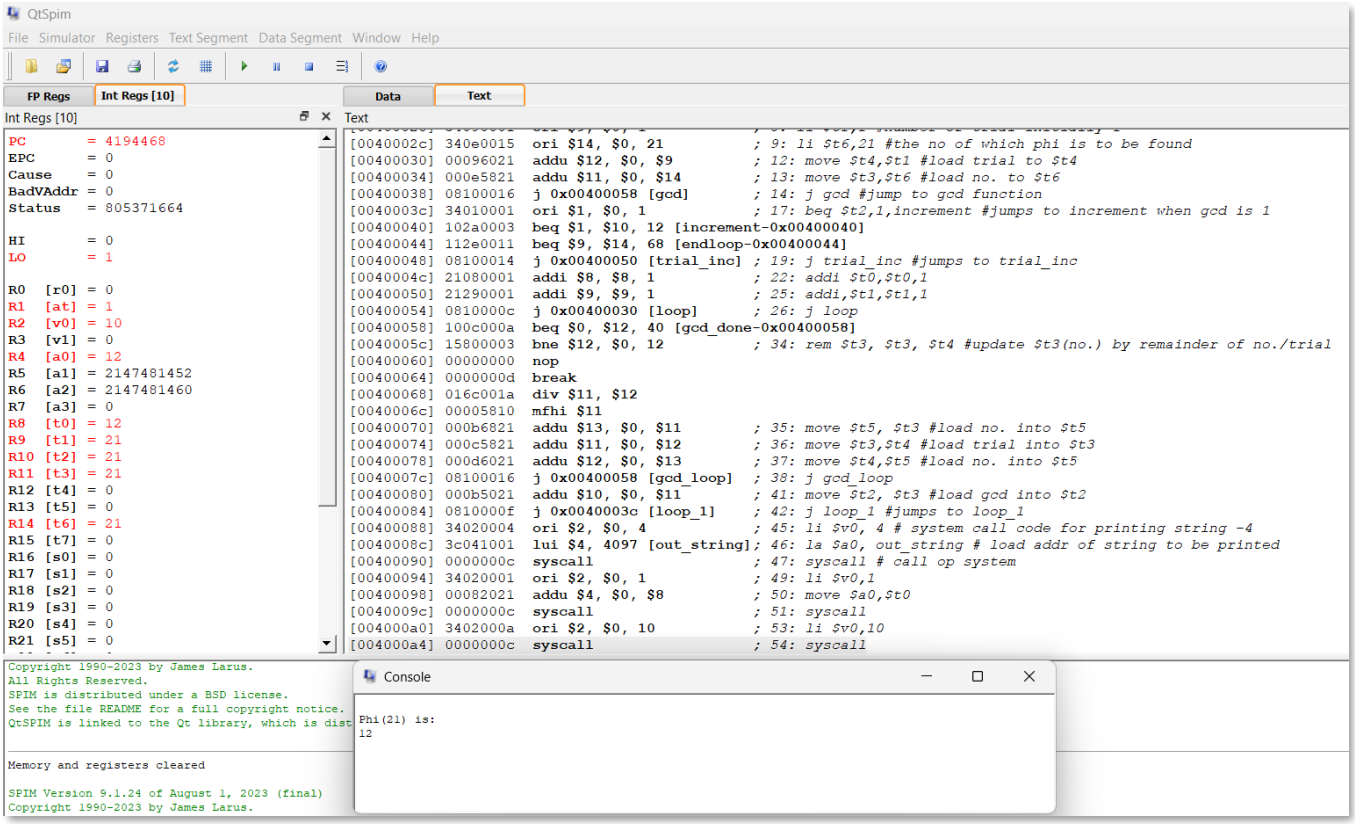
phi(1)=1, (gcd(1,1)=1)
phi(2)=1, (gcd(1,2)=1, but gcd(2,2)=2)
phi(3)=2, (gcd(1,3)=1, gcd(2,3)=1, gcd(3,3)=3)
phi(4)=2, (gcd(1,4)=1, gcd(2,4)=2, gcd(3,4)=1, gcd(4,4)=4)
phi(5)=4, (gcd(1,5)=1, gcd(2,5)=1, gcd(3,5)=1, gcd(4,5)=1, gcd(5,5)=5)

(Hint: The logic for your code would be as follows

```
phi = 0;
trial = 1;
while ( trial < N)          #where N is the number under consideration (given : 21)
{
   if ( gcd(N,trial) == 1 ) phi++;
} )
```

The computed value, which is phi(21), should be printed on the screen.

Output screenshot:

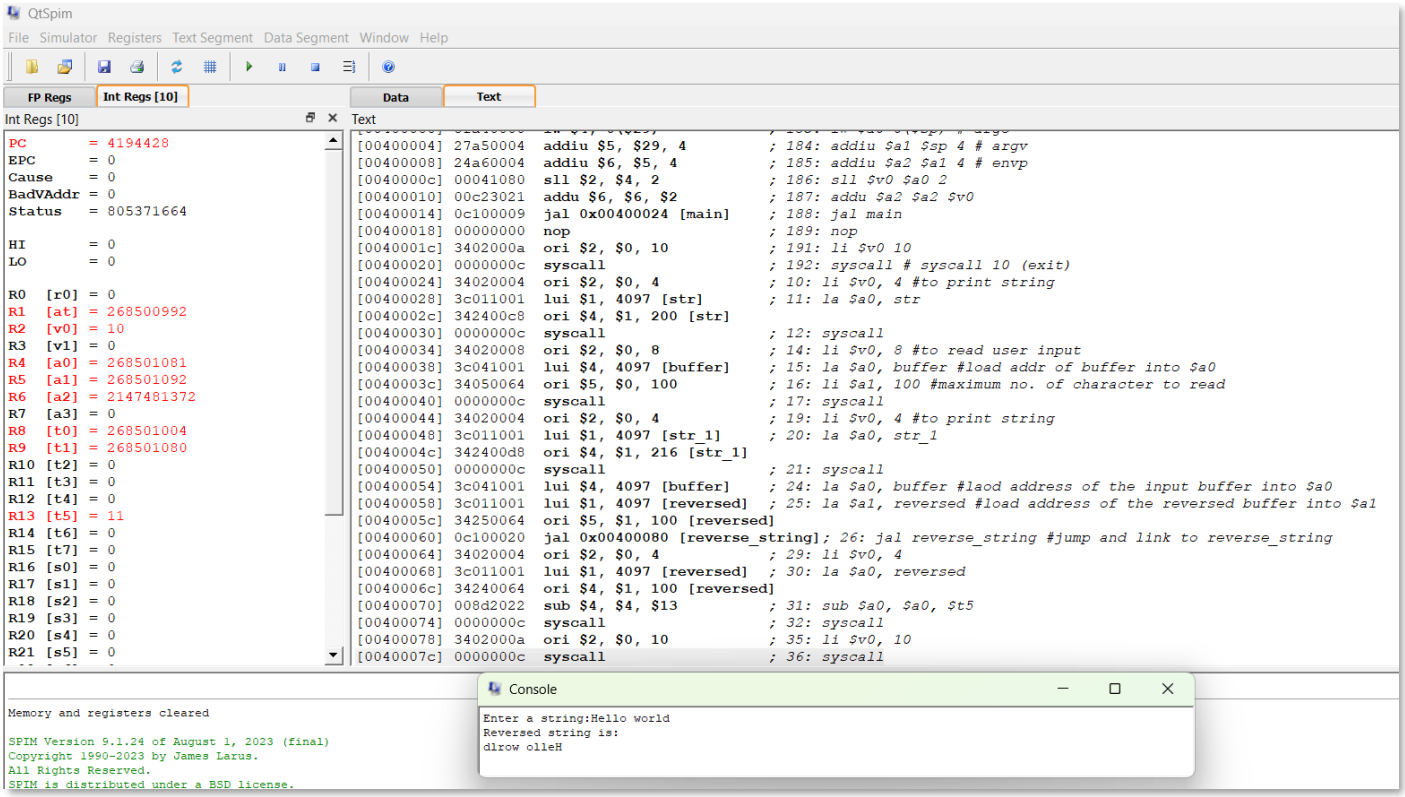


LAB 3:

Question 1:

Reverse a string entered by user. (Hint: Ask user to enter a string. After reading the string in a buffer, copy it in reversed order to a second buffer. Write out the reversed string.)

Output screenshot:

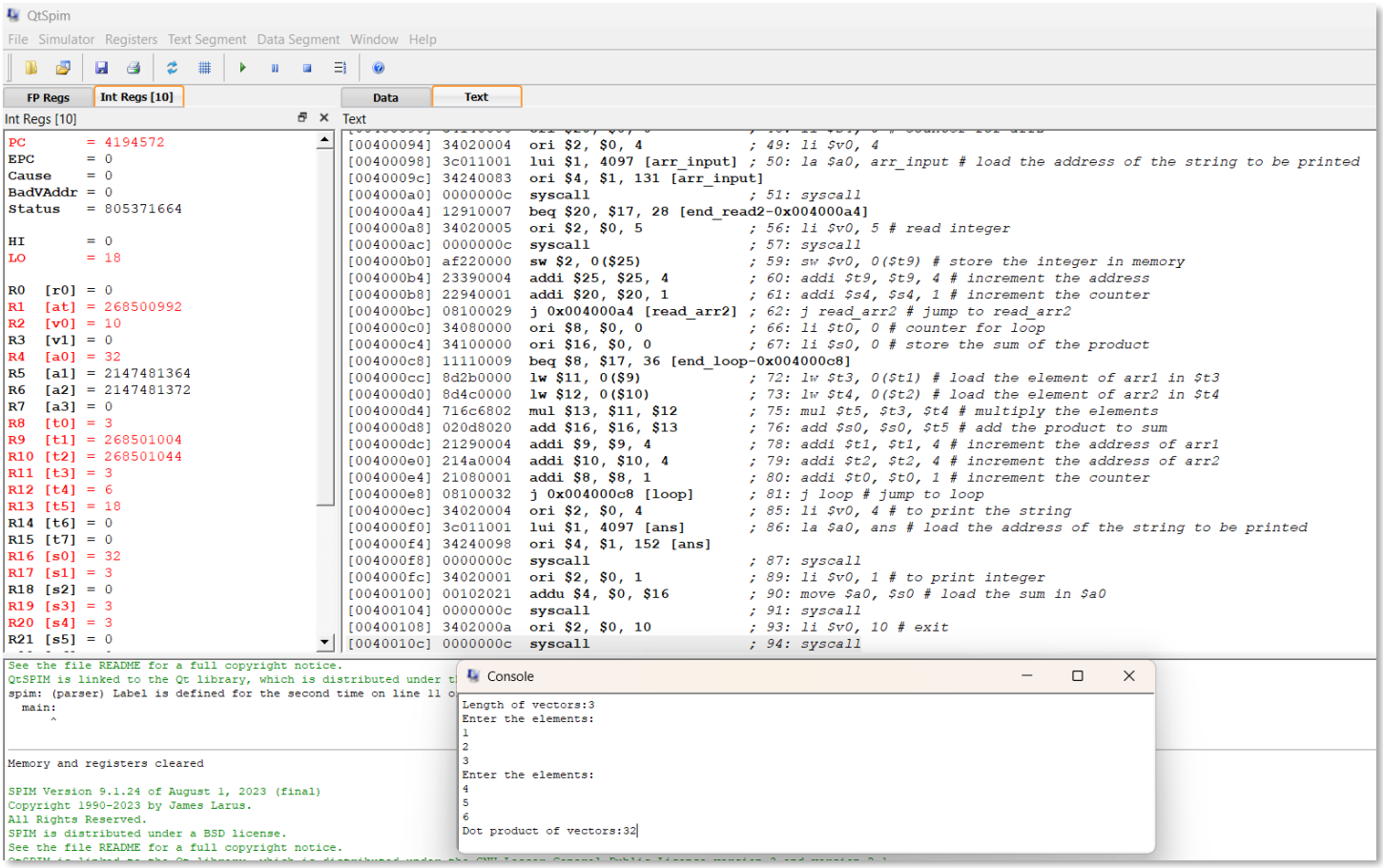


Question 2:

Compute the dot product of two vectors each of length 5. Ask the user to enter the value of each element of the two vectors. Display the dot product.

(Hint: The dot product of two vectors is sum of product of the corresponding elements. For example, (1,2,3) dot (4,5,6) is 1*4+2*5+3*6 = 32)

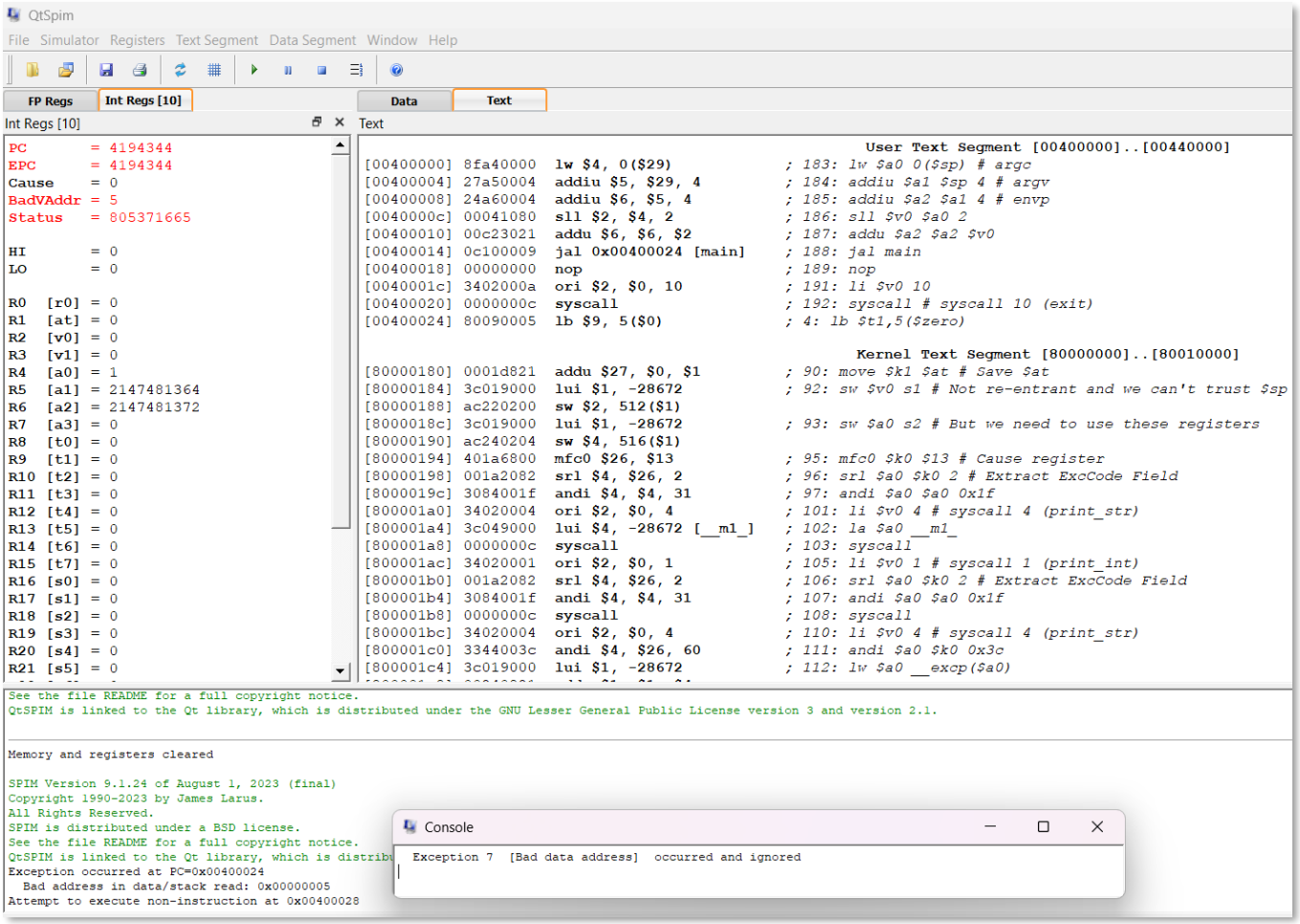
Output screenshot:



Question 3:

Use lb $t1, 5($zero) to cause an exception when attempting to load a byte from address 5. What is the address of the lb instruction in your program? What is the value of the cause register, the exception code, the vaddr, and the epc when the exception occurs?

Output screenshot:



Ans:

Cause register is 0.

Exception code is 7.

Vaddr is 5.

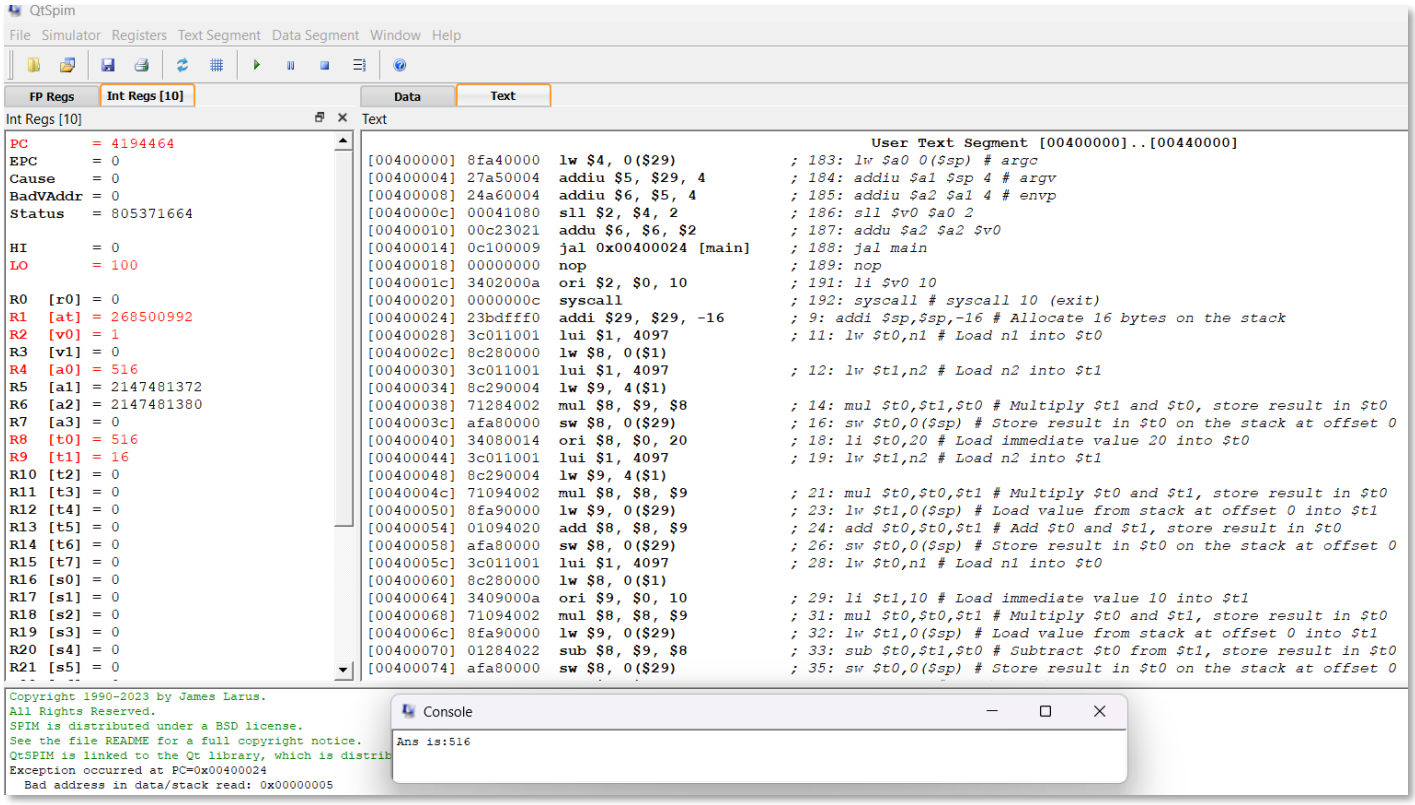
Epc when exception occurred is 4194344.

LAB 4:

Question 1:

Evaluate the expression ‘ab-10a+20b+16’. Consider that only $t0 and $t1 are available to store temporary values. Store a=10 and b=20 in data section. Use stack for other memory requirements. Display the sum.

Output screenshot:



Question 2:

Find the maximum of the three expressions: x*x ; x*y ; y*5. Take x and y as input from user. Write a global subroutine, in another file, to calculate values of these expressions. Write a subroutine to find maximum of two integers and use it to find the maximum of these three expressions. Display the result.

Output screenshot: