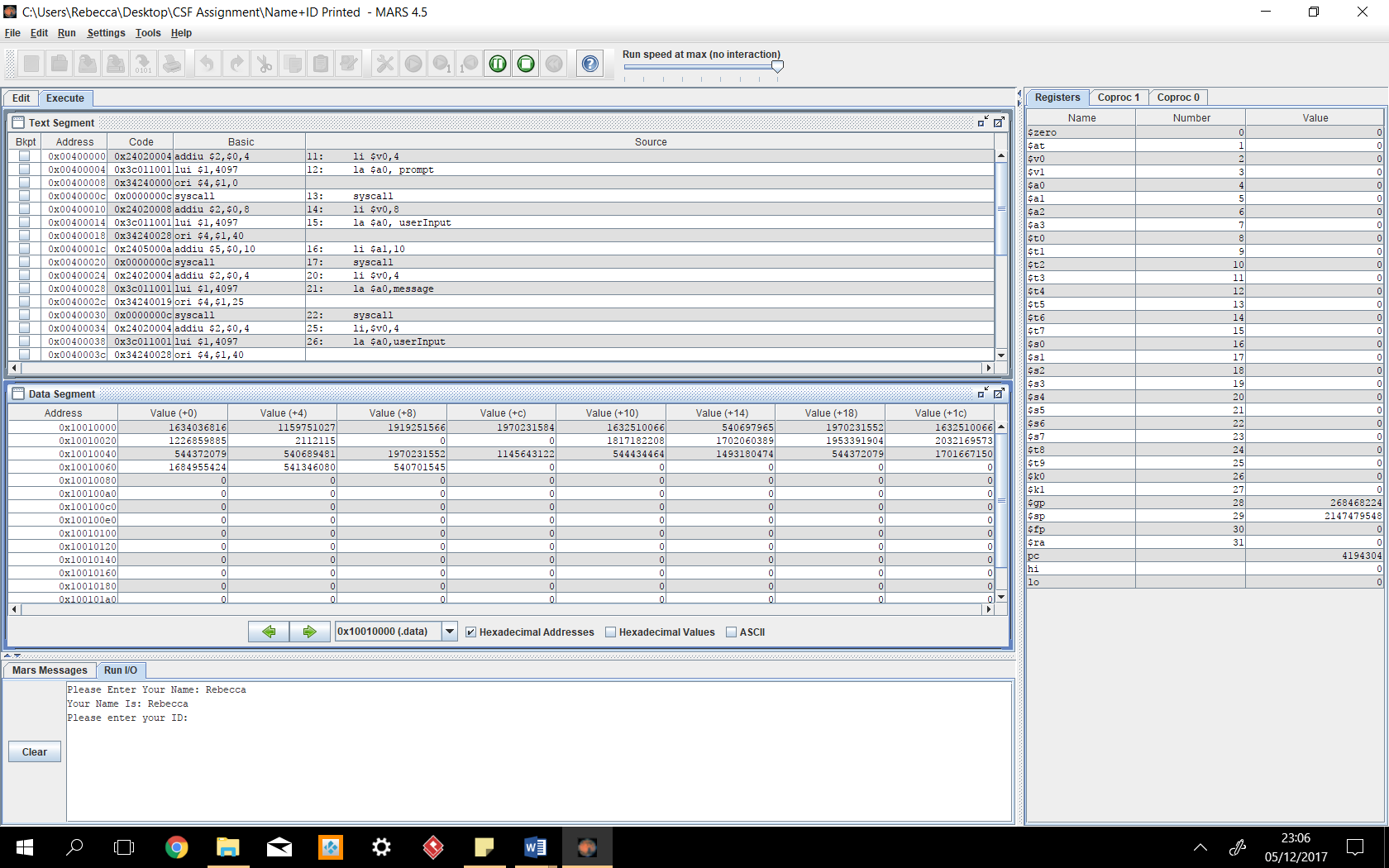
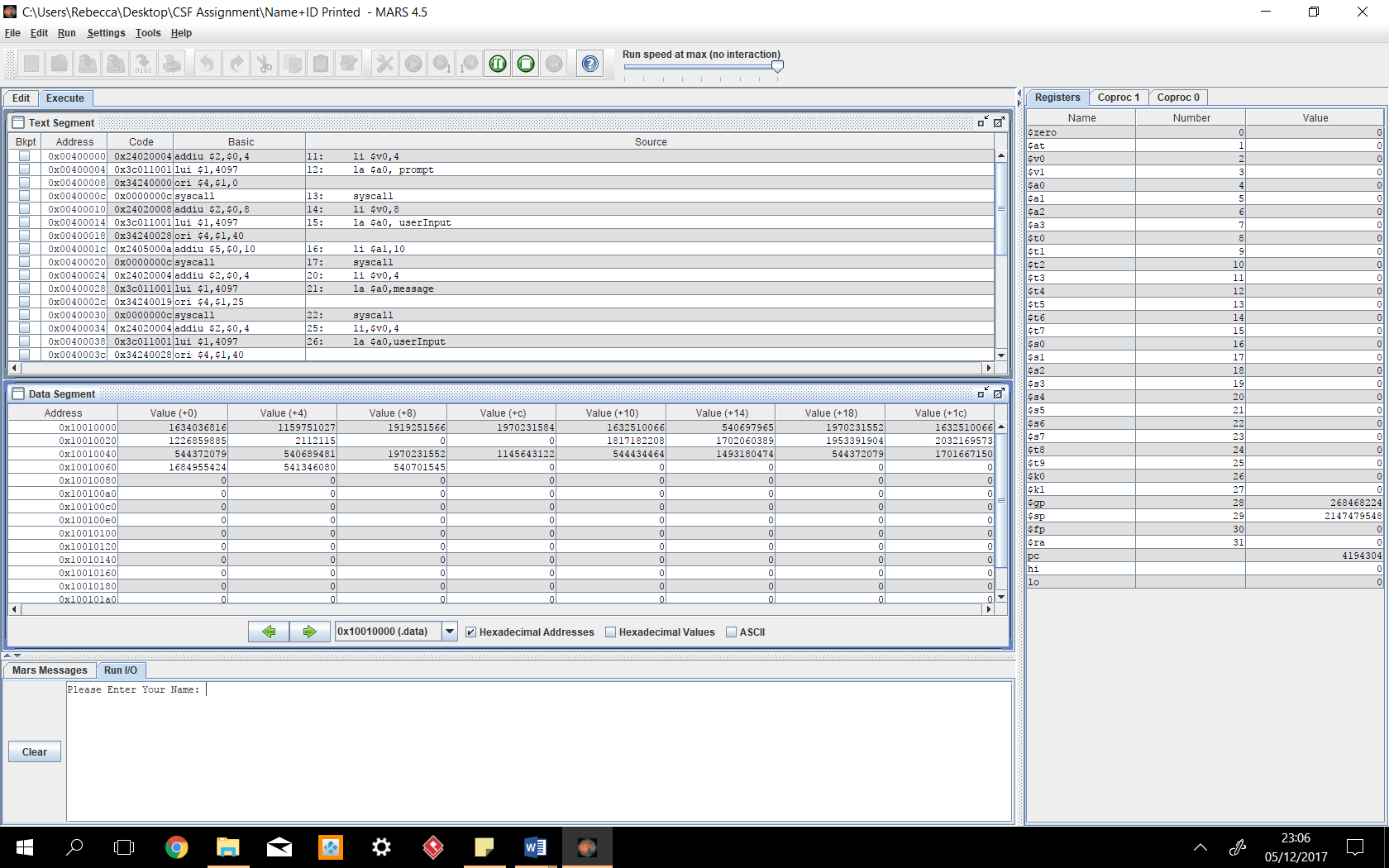
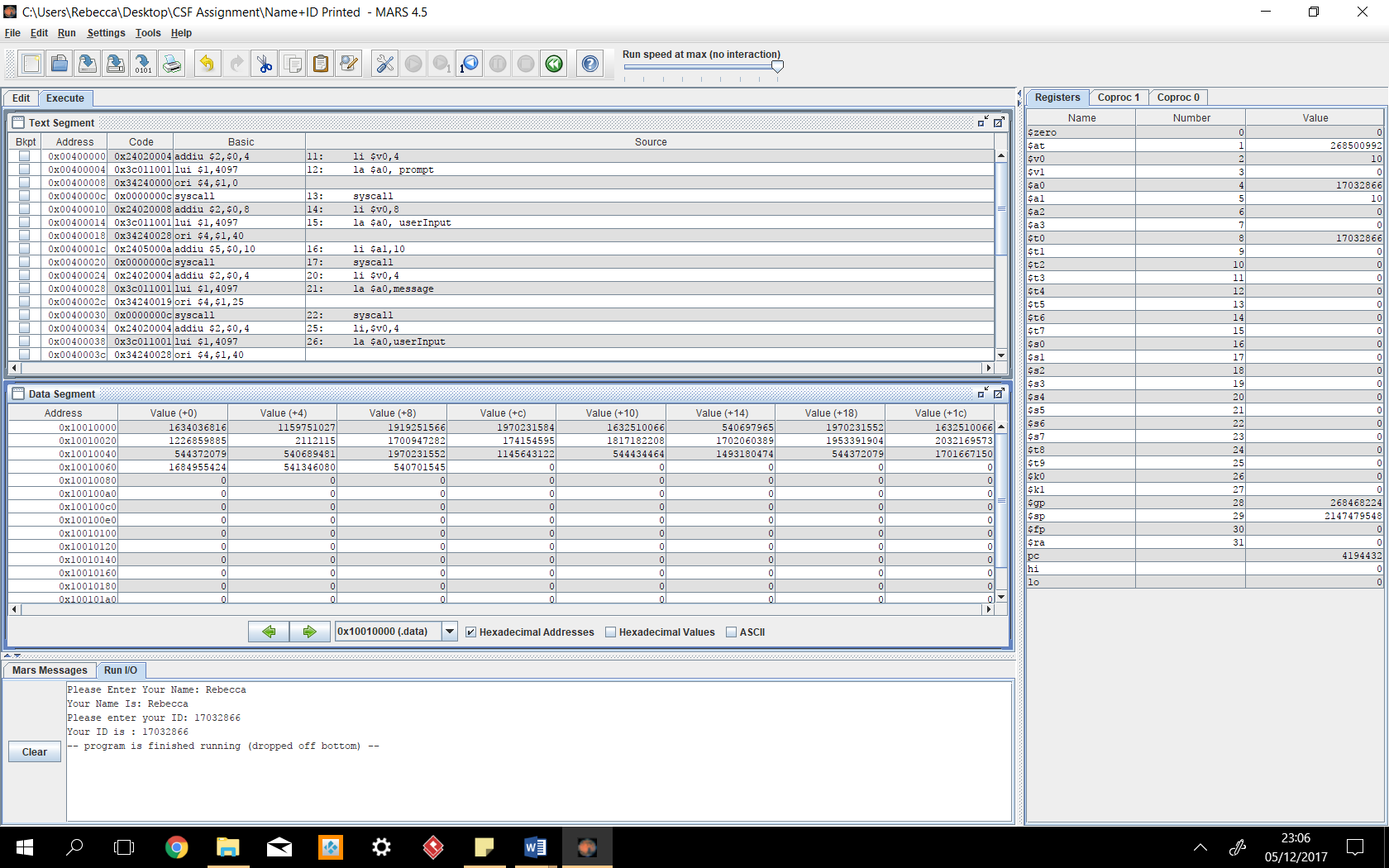
## Computer Systems Fundamentals

**MIPs Assembler**

**By Rebecca Clarke 17032866**

**Task A**

My program works by displaying a message to the user, asking for the input as a string and when the program has received an input it then displays another message and the string together. Example: Please Enter Your Name: (Message 1) Your Name Is (Message 2): Rebecca (Input) The other part of my program works by displaying another message to the user asking for the input as an integer and displaying another message and the integer together. Example: Please Enter Your ID: (message 1) Your ID is (message 2) 17032866 (Input.)

.data #Items underneath stored in the .data registery

prompt: .asciiz "Please Enter Your Name: "

message: .asciiz "Your Name Is: "

userInput: .space 10

prompt1: .asciiz "Please enter your ID: "

message1: .asciiz "Your ID is : "

finalmessage: .asciiz "Your Name and ID is: "

.text #used to keep the actual code in

main:

#getting Users input as text

li $v0,4 #prints string

la $a0, prompt #address of the print string

syscall #Calls the above code to be carried out

li $v0,8 #Reads string

la $a0, userInput #address of the input buffer

li $a1,10 #sets the maximum number of characters that the program can read to 10

syscall #calls the above code to be carried out

#display your name is

li $v0,4 #Loads the string within the $ao register and then prints it

la $a0,message #Load the message in $a0

syscall #Calls the above code to be carried out

#displays the name of the user input

li,$v0,4 #prints string

la $a0,userInput #address of the print string

syscall #Calls the above code to be carried out

#prompt the user to enter number

li $v0,4 #prints string

la $a0, prompt1 #address of the print string

syscall #Calls the above code to be carried out

#get uset input //Reading Integer

li $v0,5 #Loads the int within the $ao register

syscall #Calls the above code to be carried out

#store result in $t0

move $t0,$v0 #moves the $to registery to $vo registery

#display message

li $v0,4 #prints string

la $a0,message1 #address of the print string

syscall #Calls the above code to be carried out

#print or display number

li $v0,1 #loads the int in $v0 and prints it

move $a0,$t0 #moves the $a0 registery info to $st0 registery

syscall #Calls the above code to be carried out

#Ends the program

li $v0,10 #Makes the program exit the function main:

**Task B**

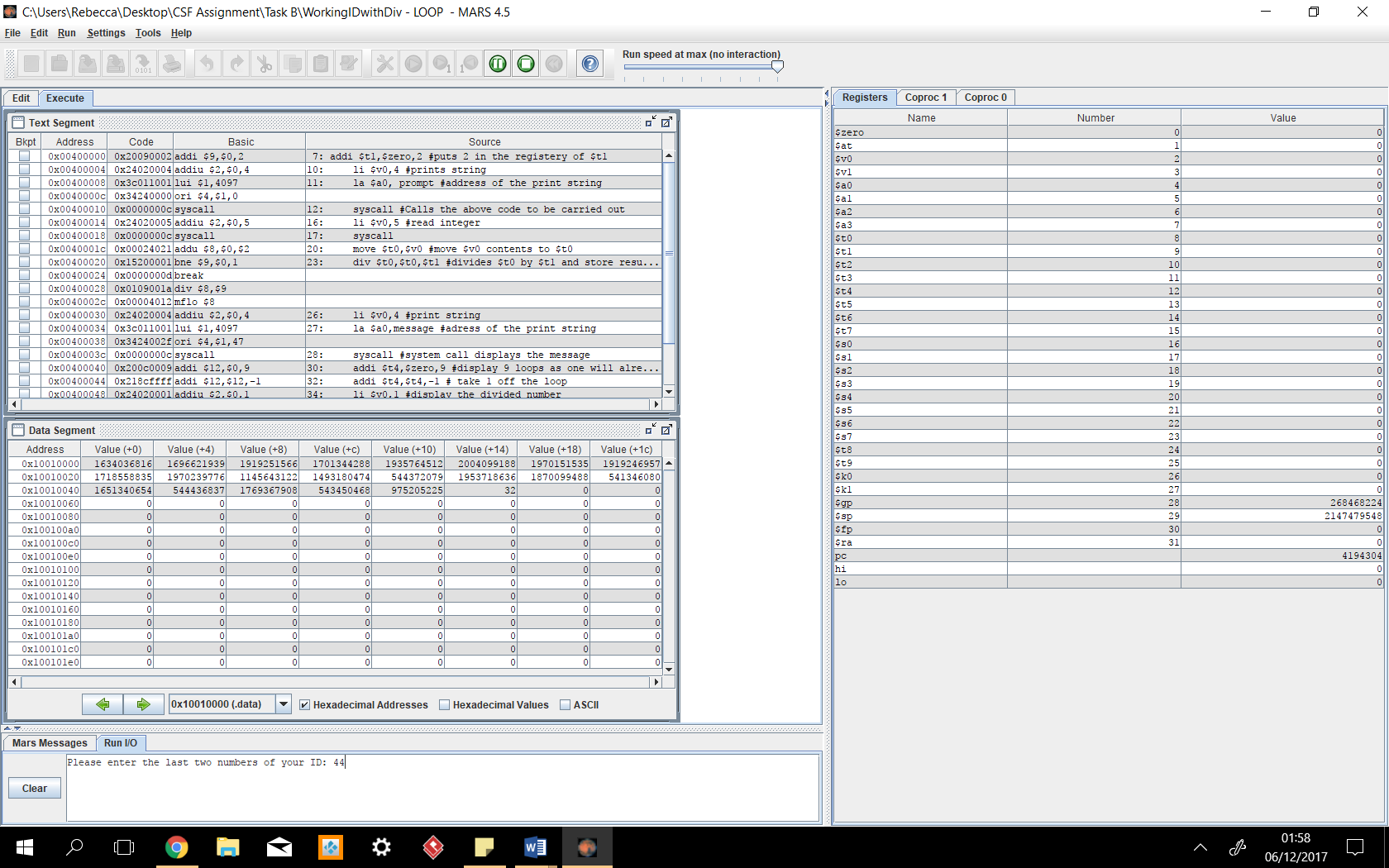
This program prompts the user to input the last two digits of their ID number, the program then takes this number as the user input, it then divides this number by 2, once it has divided it by 2 it then loops the result 10 times and then displays a message saying the last two ID numbers divided by 2 and looped 10 times is: then it displays the result after this message.

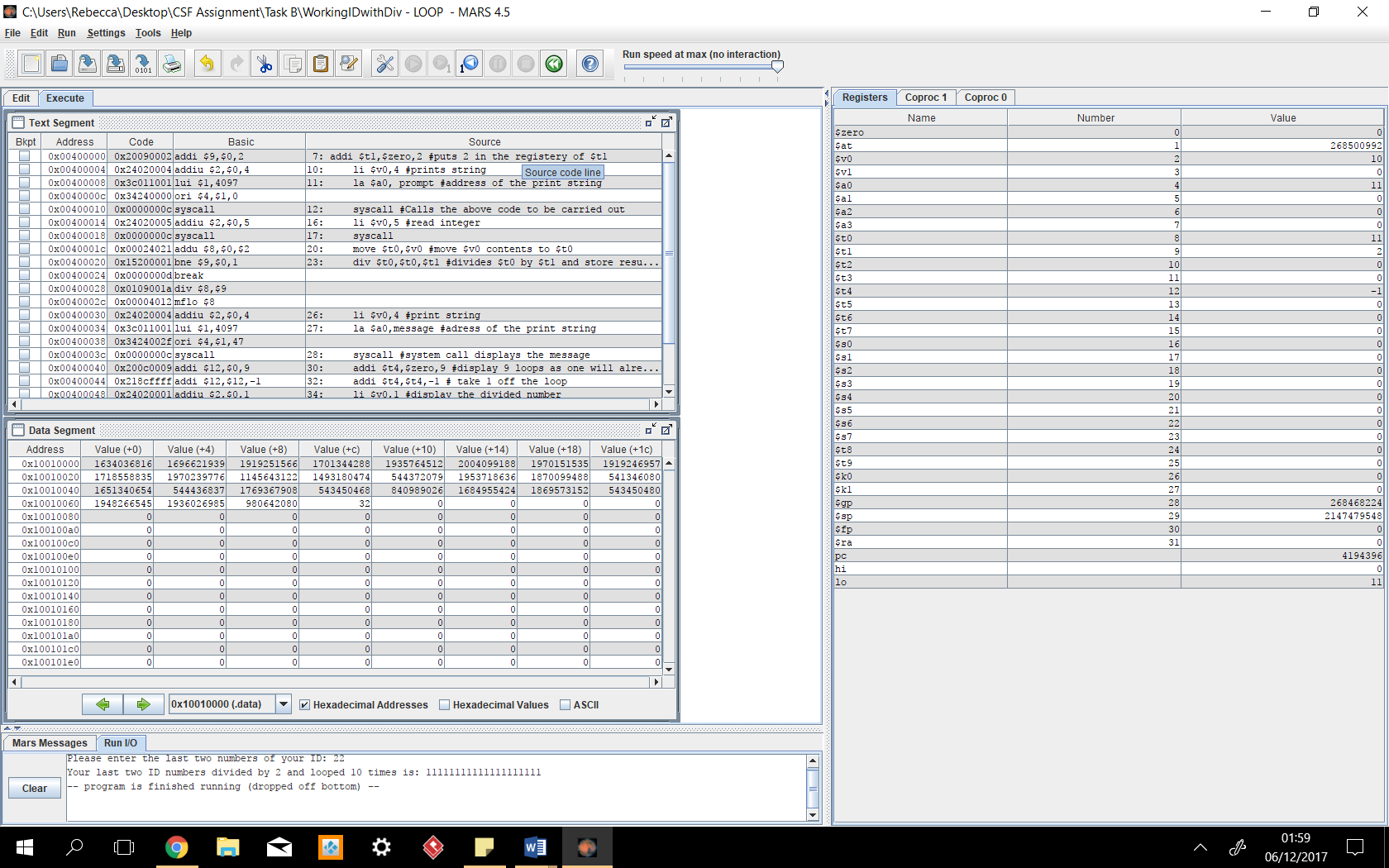
For example:

User Input: 44

Divided by 2: 22

Looped 10 times: 22222222222222222222





.data

prompt: .asciiz "Please enter the last two numbers of your ID: "

message: .asciiz "Your last two ID numbers divided by 2 and looped 10 times is: "

.text

main:

#sets the $t1 registery to 2

addi $t1,$zero,2 #puts 2 in the registery of $t1

#prompt the user to enter number

li $v0,4 #prints string

la $a0, prompt #address of the print string

syscall #Calls the above code to be carried out

#get uset input //Reading Integer

li $v0,5 #read integer

syscall

#store result in $t0

move $t0,$v0 #move $v0 contents to $t0

#double result in $t0

div $t0,$t0,$t1 #divides $t0 by $t1 and store result in $t0

#display message

li $v0,4 #print string

la $a0,message #adress of the print string

syscall #system call displays the message

addi $t4,$zero,9 #display 9 loops as one will already be displayed

loop: #initalise the loop

addi $t4,$t4,-1 # take 1 off the loop

#print or display number

li $v0,1 #display the divided number

move $a0,$t0 #move contents of $t0 to $a0

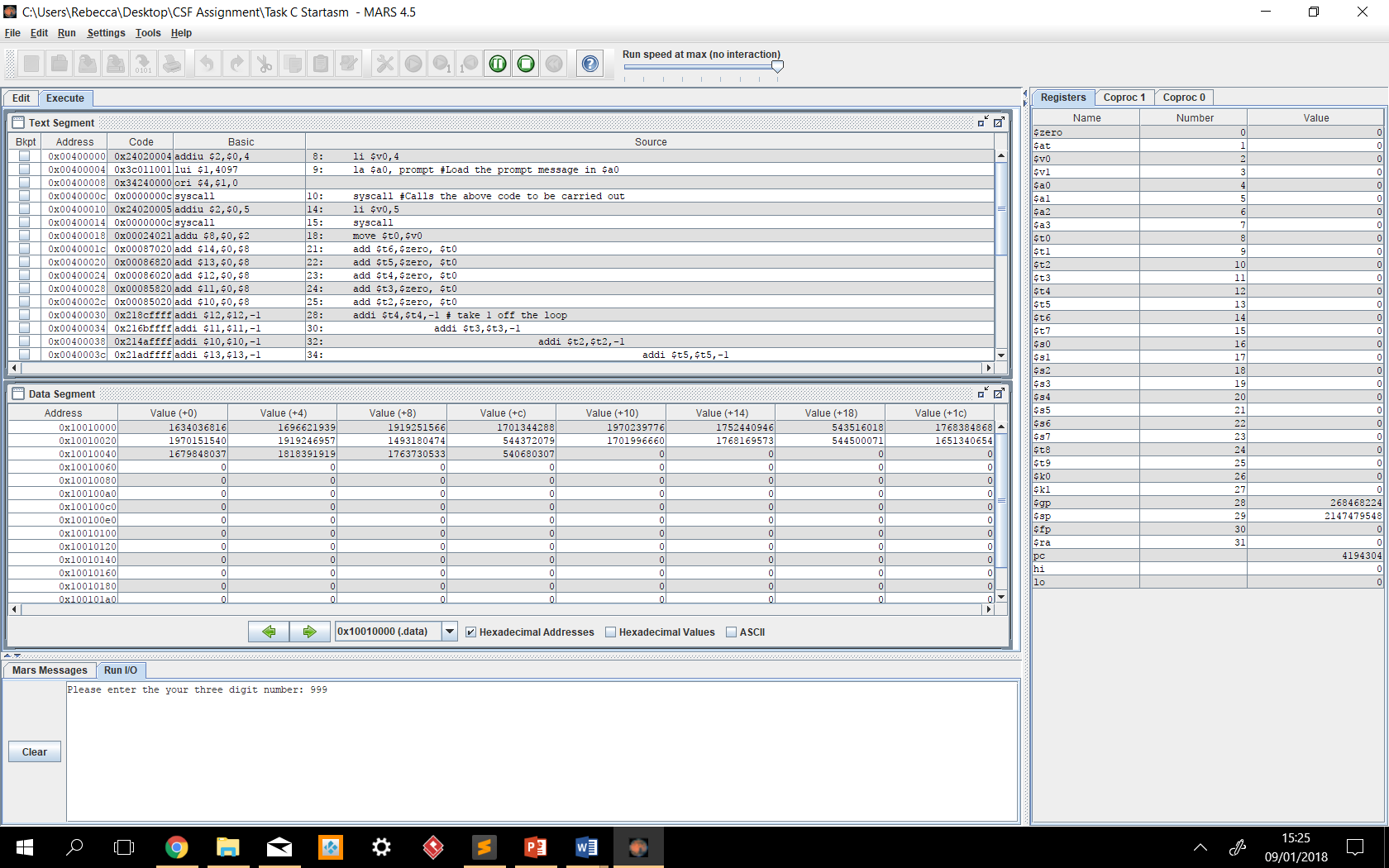
syscall #system call displays divided result

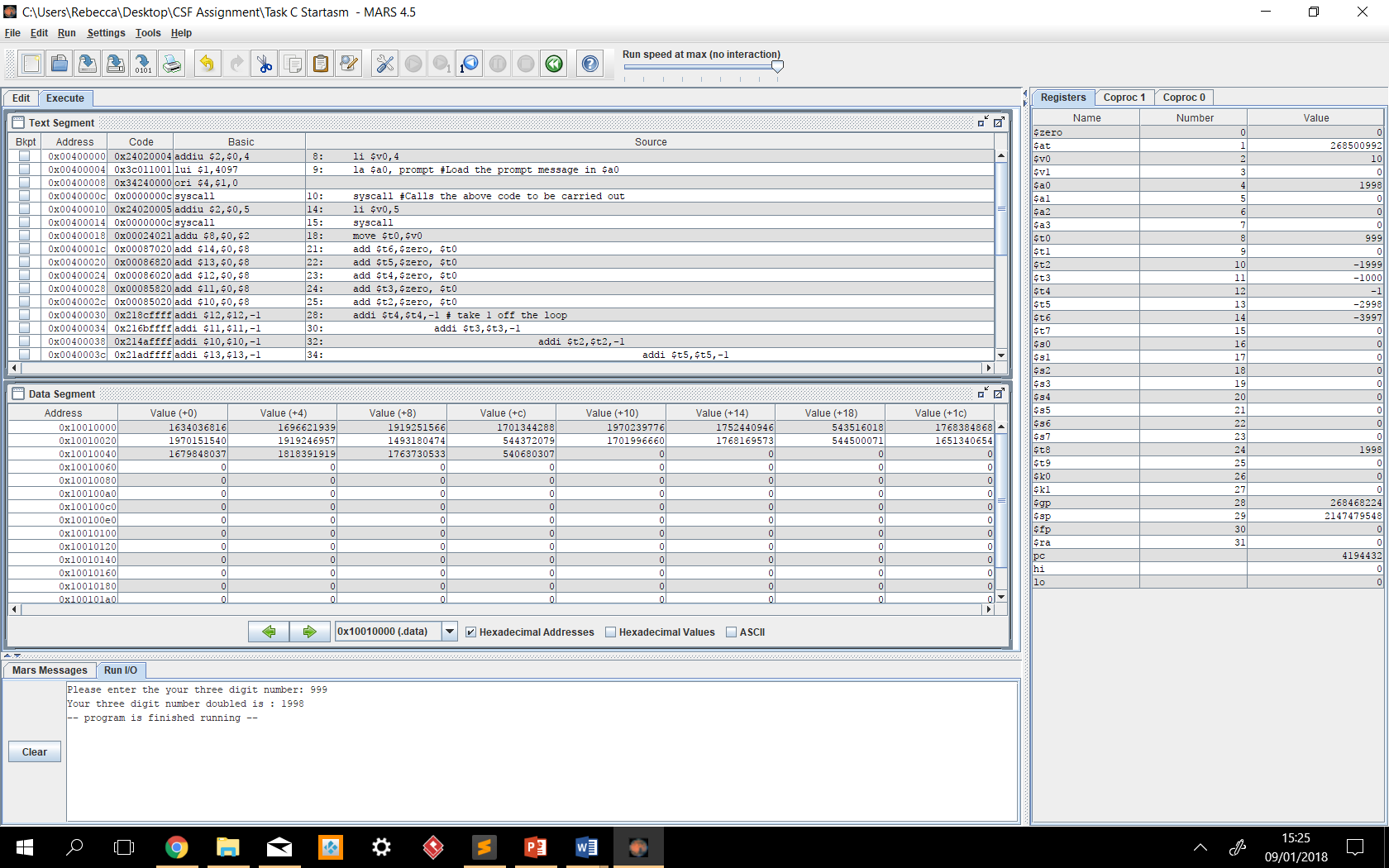
bgez $t4,loop #branch loop if greater than 1 or equal to 0

li $v0,10 #Makes the program exit the function main

**Task C**

This program works by taking the users input of a three-digit number and adds this number to the same three digit number which the user inputted. This addition takes place within a nested loop to increase the time which the calculation takes. This is program would take longer or slower to complete the calculation depending on what three digit number was inputted. My program was unsuccessful, and this may be of one of two reasons. The first reason is that my computer can handle a lot of processes very quickly meaning it Is completing the tasks too quickly and I haven’t got enough nested loops or the other reason is that it may not be storing the number within the directory of $t0 and then adding zero to the users inputted number and storing this within another directory to then process to be taken away 1 within the loop. My program uses 4 nested loops and takes the users number, takes 1 off this number and stores this back into the original directory so it can carry on doing this loop before moving onto the next loop. These nested loops are supposed to increase the time it takes the computer to carry out the operation as it must carry out that loop before moving onto the next loop.





.data

prompt: .asciiz "Please enter the your three digit number: "

message: .asciiz "Your three digit number doubled is : "

.text

main:

#prompt the user to enter number

li $v0,4 #loading the immediate

la $a0, prompt #Load the prompt message in $a0

syscall #Calls the above code to be carried out

#get uset input //Reading Integer

li $v0,5

syscall

#store result in $t0

move $t0,$v0

#loop setup numbers

#stores the users number within another directory and adds 0 to it

add $t6,$zero, $t0

add $t5,$zero, $t0

add $t4,$zero, $t0

add $t3,$zero, $t0

add $t2,$zero, $t0

loop: #initalise the loop

addi $t4,$t4,-1 # take 1 off the loop

loop1: #initalise the loop

addi $t3,$t3,-1

loop2: #initalises the loop

addi $t2,$t2,-1

loop3:

addi $t5,$t5,-1

loop4:

addi $t6,$t6,-1

add $t8,$t0,$t0

bgez $t6,loop4

bgez $t5,loop3

bgez $t2,loop2

bgez $t3,loop1 #branch loop if greater than 1 or equal to 0

bgez $t4,loop

#display message

li $v0,4

la $a0,message

syscall

#print or display number

li $v0,1

la $a0,($t8)

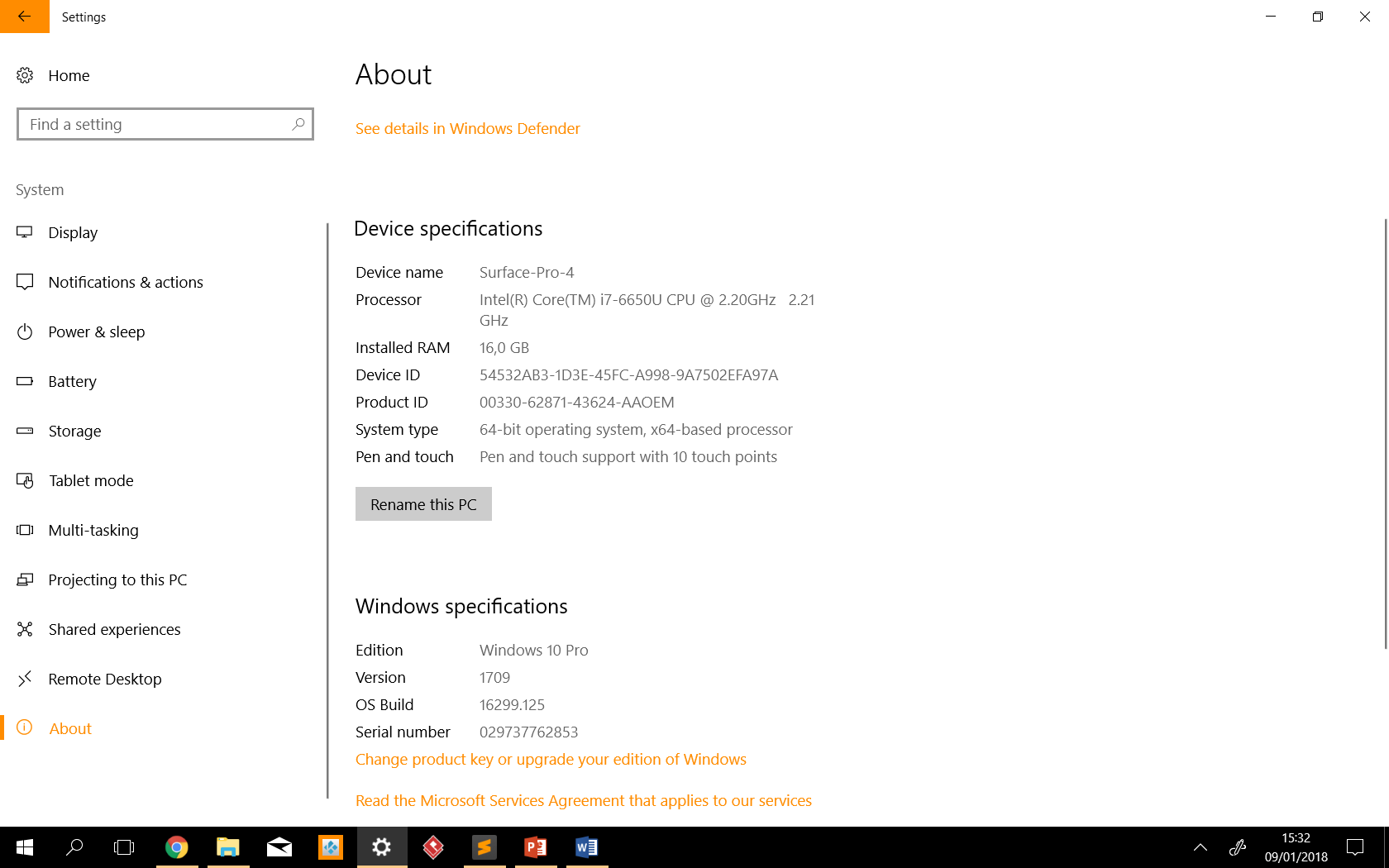
syscall

#Makes the program exit the function main

li $v0,10

syscall

**System Details**



Input

Gets The user input of the three digit number.

Starts the loops

Carries out the loops within the nest, once one loop finishes it moves onto the next until the most inner loop. This loop has the calculation within it and carries out this operation.

Shows user the calculated number

Displays the message containing calculated number