**DEBUGGING USING BREAKPOINTS, SINGLE STEP, USE OF LOOPS & CONDITIONAL STATEMENTS**

**LAB # 04**



**Fall 2022**

**CSE304L Computer Organization & Architecture**

**Submitted by:**

Maaz Habib[20PWCSE1952]

Hamid Ur Rehman [20PWCSE1969]

Danyal Khursheed [20PWCSE1959]

**Class Section:**

C

**Submitted to:**

Dr. Amaad Khalil

**Department of Computer Systems Engineering**

**University of Engineering and Technology, Peshawar**

**DEBUGGING USING BREAKPOINTS:**

Debugging using breakpoints in SPIM is a process of setting points in your code where execution will temporarily halt. This allows you to inspect the state of the program and make sure it is running as expected.

**Set a breakpoint:**

To set a breakpoint, right-click on the line of code you want to debug, and select "Toggle Breakpoint". A red dot will appear to the left of the line, indicating that a breakpoint has been set.

**Task-1.**

**Code:**

.data

str1: .asciiz"The division is: "

str2: .asciiz"\n The Remender is: "

str3: .asciiz"\n Multiplication is: "

.text

main:

addi $t0, $0, 40

addi $t1, $0, 6

mult $t0, $t1

mflo $t6

div $t0, $t1

li $v0,4

la $a0,str1

syscall

li $v0,1

mflo $a0

syscall

li $v0,4

la $a0,str2

syscall

li $v0,1

mfhi $a0

syscall

li $v0,4

la $a0,str3

syscall

li $v0,1

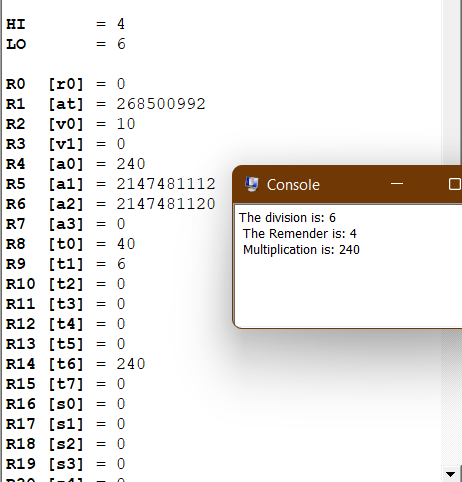
move $a0,$t6

syscall

li $v0,10

syscall

**Output:**



**Task-2:**

**Code:**

.data

str1: .asciiz"The Right Shift is: "

str2: .asciiz"\n The left Shift is: "

.text

main:

addi $t0, $0, 20

li $v0,4

la $a0,str1

syscall

srl $a0, $t0, 3

li $v0,1

syscall

li $v0,4

la $a0,str2

syscall

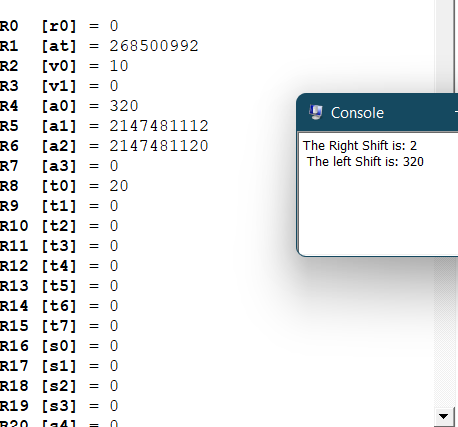
sll $a0, $t0, 4

li $v0,1

syscall

fini: jr $ra

**Output:**



**Task-3:**

**Code:**

.data

str1: .asciiz"Please enter num a: "

str2: .asciiz"\n The result in the form 18a: "

.text

main:

li $v0,4

la $a0,str1

syscall

li $v0,5

syscall

move $t0,$v0

sll $t1,$t0,4

sll $t2,$t0,1

add $t3,$t1,$t2

li $v0,4

la $a0,str2

syscall

li $v0,1

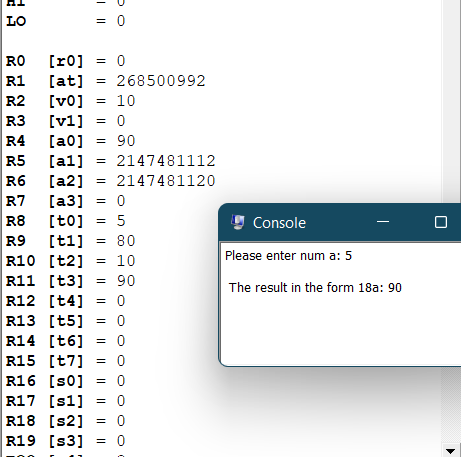
move $a0,$t3

syscall

li $v0,10

syscall

**output:**

****

**USE OF LOOPS & CONDITIONAL STATEMENTS:**

**Loops:**

Loops in MIPS assembly are used to repeat a block of code multiple times. The most commonly used loop construct is the "for" loop. A basic "for" loop in MIPS assembly

**Task-4:**

Write a program that take two integer and if 1 is input it will display their sum and if 2 is input it will display their Subtraction.

**Code:**

.data

str1: .asciiz"Enter first number: "

str2: .asciiz"\n Enter second number: "

str3: .asciiz"\n press 1 for addition or 2 for subtraction: "

str4: .asciiz"The final output is: "

.text

main:

li $v0,4

la $a0,str1

syscall

li $v0,5

syscall

move $t0,$v0

li $v0,4

la $a0,str2

syscall

li $v0,5

syscall

move $t1,$v0

li $v0,4

la $a0,str3

syscall

li $v0,5

syscall

move $t3,$v0

li $t4,1

beq $t4,$t3 addationlabel

sub $t5,$t0,$t1

j exit

addationlabel: add $t5,$t0,$t1

exit:

li $v0,4

la $a0,str4

syscall

li $v0,1

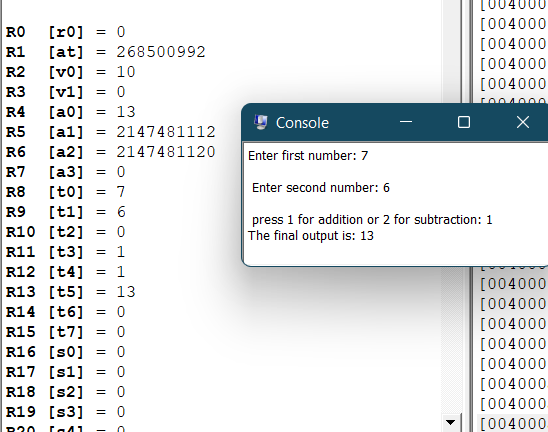
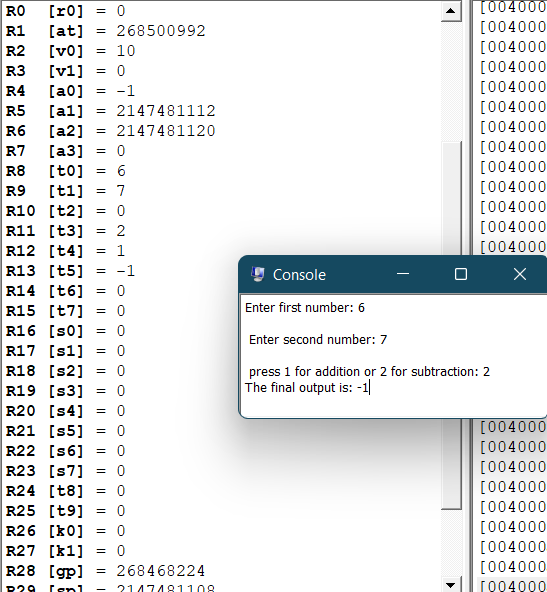
move $a0,$t5

syscall

li $v0,10

syscall

**Output:**



**Task-5:**

Write a program that take a number and then display its last digit.

**Code:**

.data

str1: .asciiz"Enter the number: "

str2: .asciiz"\n The last digit of input is: "

.text

main:

li $v0,4

la $a0,str1

syscall

li $v0,5

syscall

move $t0,$v0

li $t1,10

div $t0,$t1

li $v0,4

la $a0,str2

syscall

li $v0,1

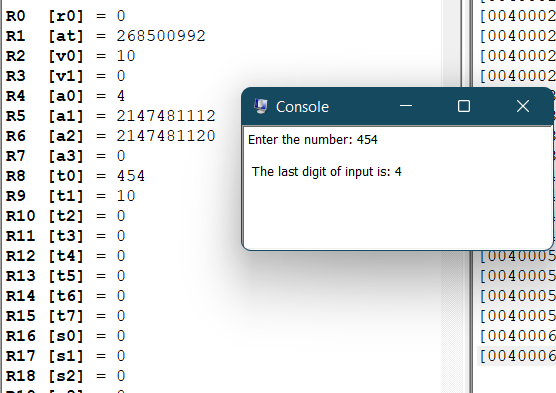
mfhi $a0

syscall

li $v0,10

syscall

**Output:**



**Task 03:**

Write a program that take a number and display its 7th bit.

**Source code:**

.data

str1: .asciiz"Enter integer of 7 bit: "

str2: .asciiz"\n The 7th bit of input integer is: "

.text

main:

li $v0,4

la $a0,str1

syscall

li $v0,5

syscall

move $t0,$v0

li $t1,0

li $t2,1

li $t3,64

and $t4,$t0,$t3

beq $t4,$t1 label

move $t5,$t2

j exit

label: move $t5,$t1

exit:

li $v0,4

la $a0,str2

syscall

li $v0,1

move $a0,$t5

syscall

li $v0,10

syscall

**Output:**

