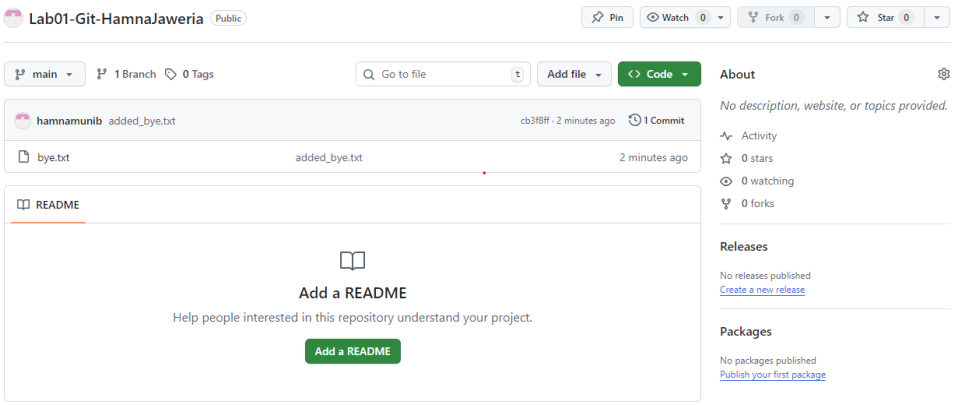
Name: Hamna Munib, Jaweria Amer	Student ID: hm09929, jc09631	section*: T3

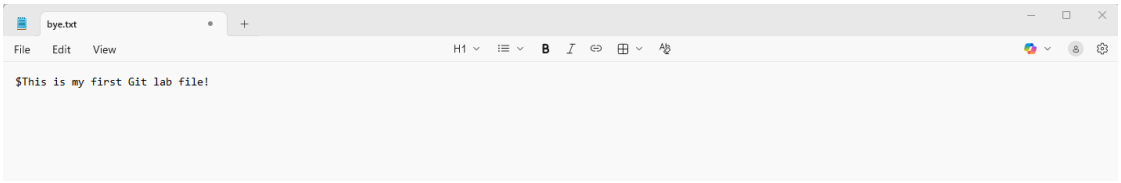
Lab 1: Getting Started with RISC-V (Assembly Language) in VS Code

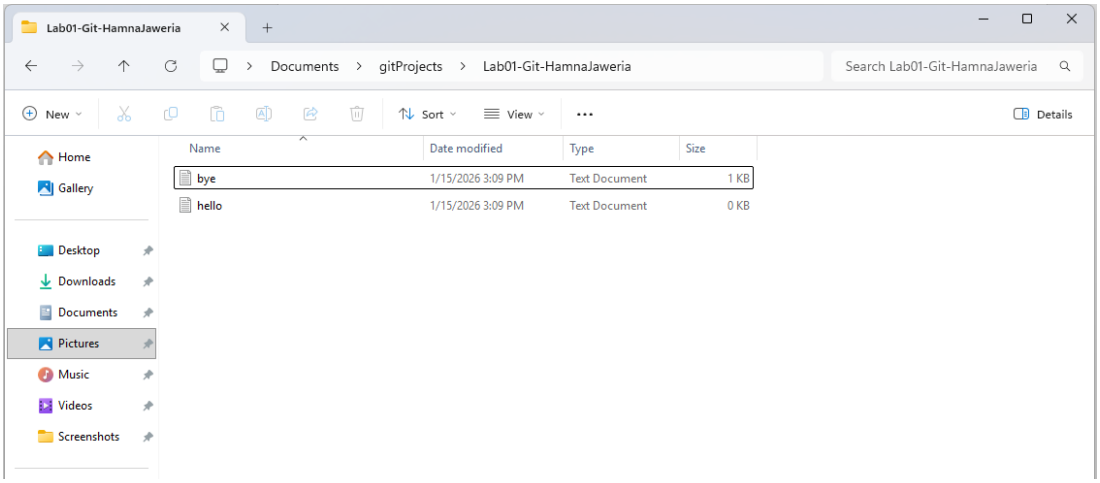
Task 1

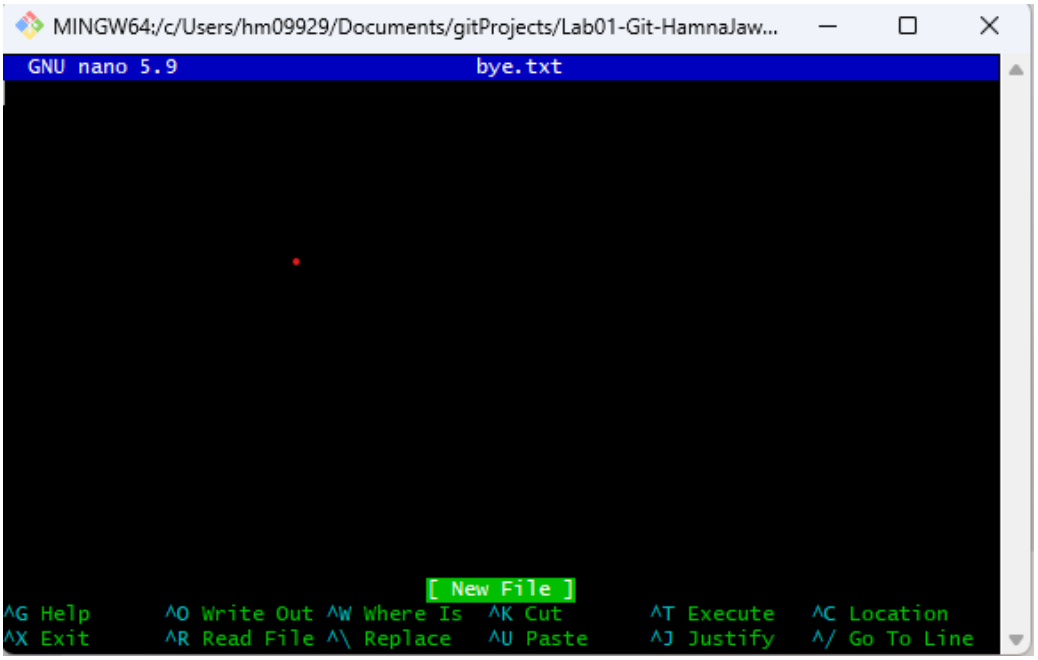
Git Repository for Lab 01



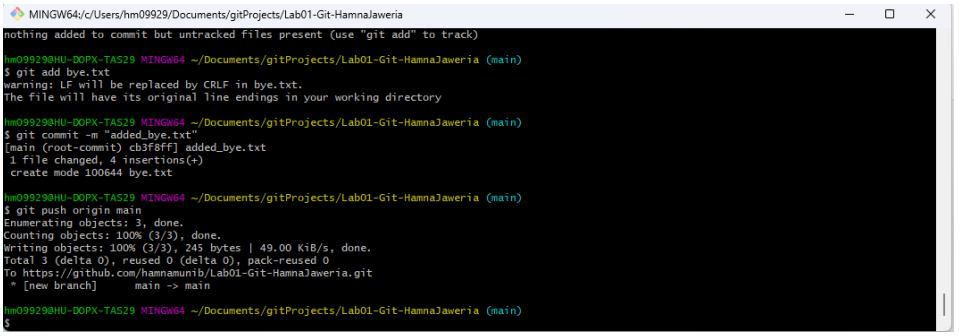
Creating and editing txt file

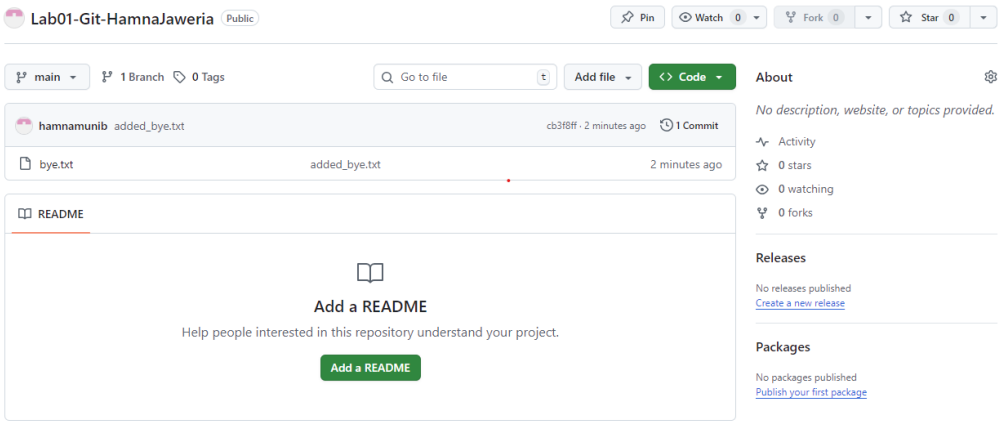




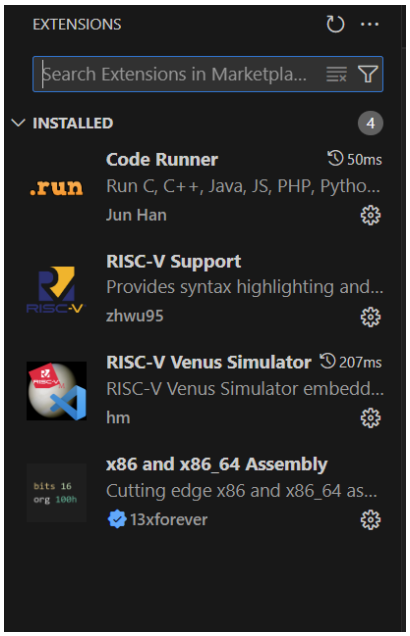


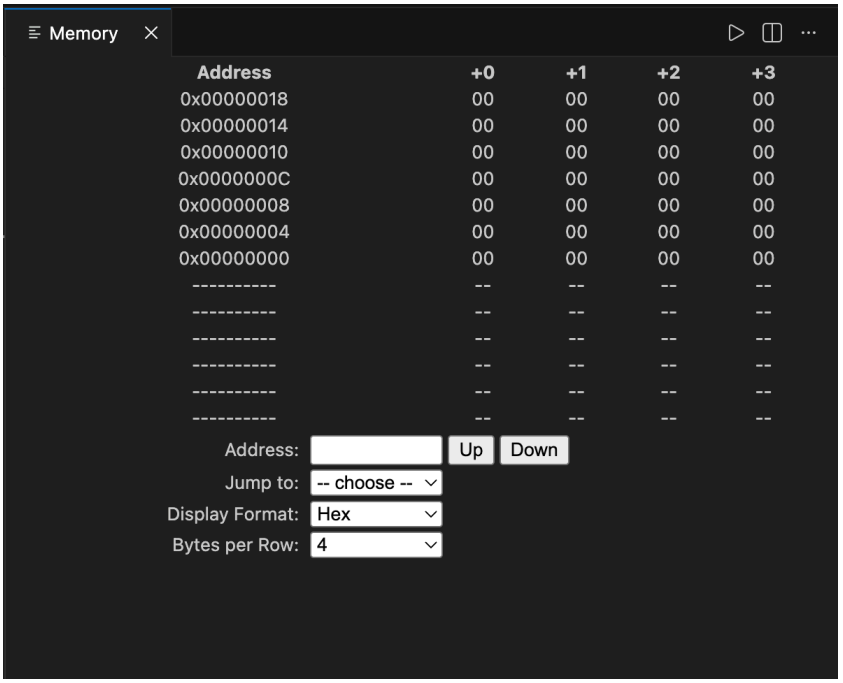
Pushing to Git

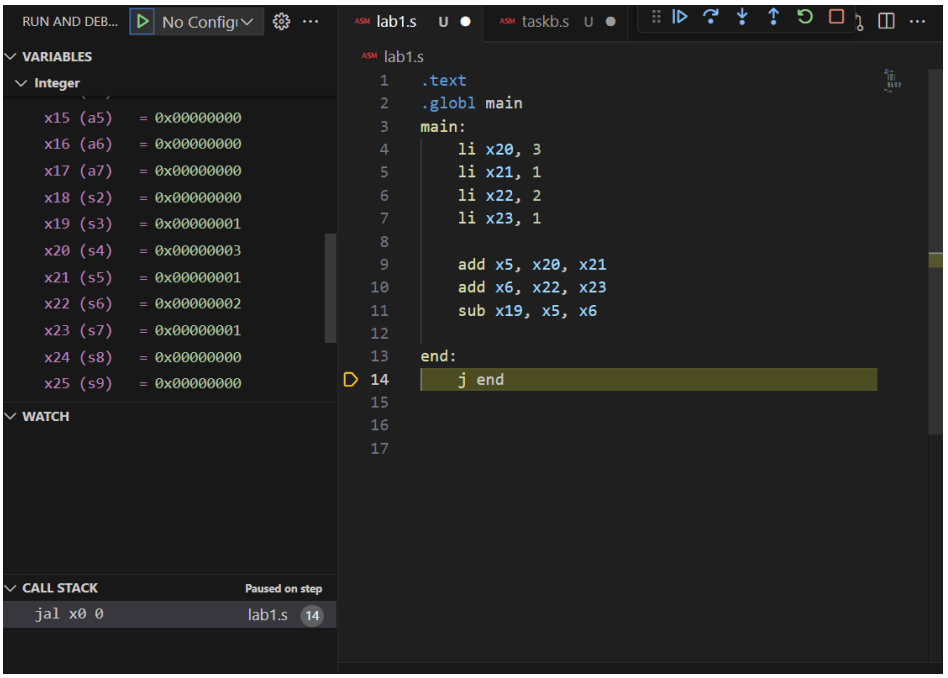




Task 02: Setting Up VS Code (RISC-V Simulation Environment)







Task 3

Convert the following statement to RISC V. You can use the same registers as given

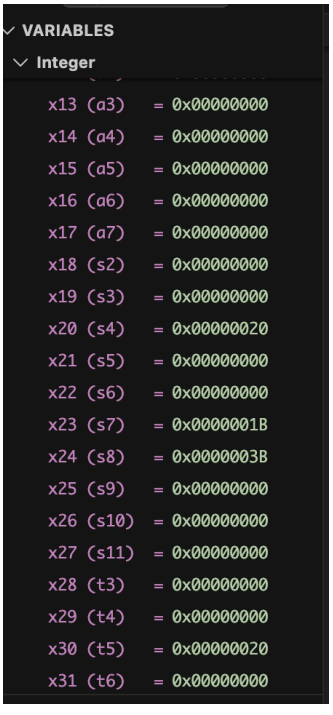
```
int a = 5;
int b = 0 + 0; a
= b + 32;
int d = (a + b) - 5;
int e = (((a - d) + (b - a)) + d);
e = a + b + d + e;
```

Code

task3.s

```
.text
.globl main
main:
    li x20, 5 # int a=5
    li x21, 0 #b
    # li x22, 0 #c
    # li x23, 0 #d
    # li x24, 0 #e

    #x01= a +b
    #x02= a -d

    addi x21, x22, 0 # int b=0+0
    addi x20, x21, 32 #int a = b+32
    add x30, x20, x21 #int random = (a+b)
    addi x23, x30, -5 # int d = random -5
    sub x2, x20, x23 #a-d
    sub x3, x21, x20 #b-a
    add x24, x2, x3 # e = (a-d) + (b-a)
    add x24, x24,x23  #e = (a-d) + (b-a) -5
    add x4, x20, x21 #random =a+b
    add x5, x23, x24 #random= d+e
    add x24, x4, x5 #e= a+b+d+e

end:
    j end
```

Output

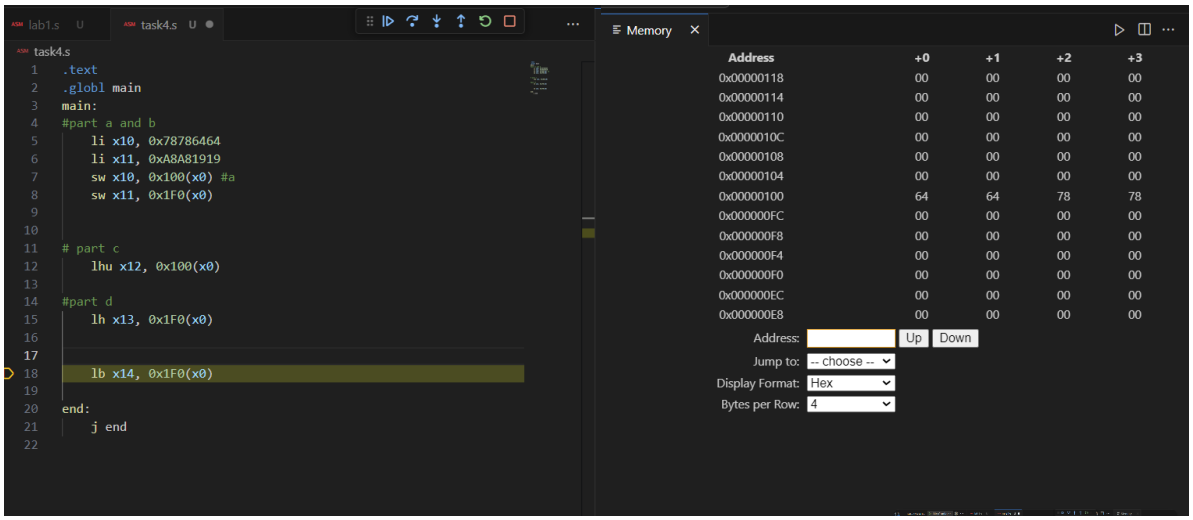


Final value of e = 3B (hex) and 59 (in decimal)
e value 3B stored in register x24

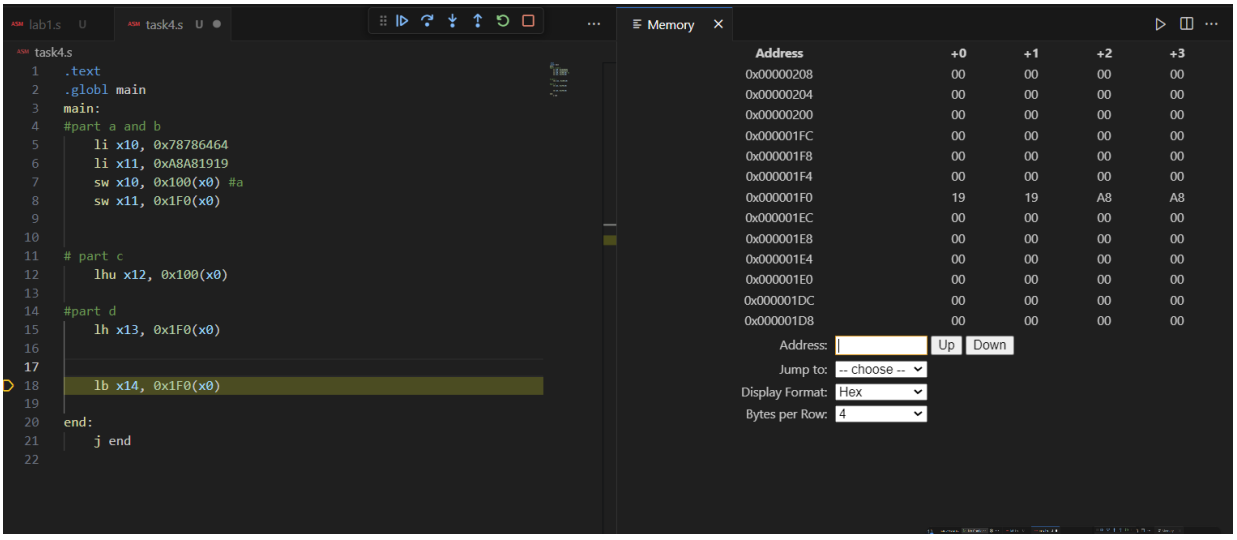
Task 4a

Initialize the register x10 and x11 with values 0x78786464, 0xA8A81919, respectively manually. Write the RISC-V assembly code for each item below. Try guessing the result in each destination before executing the instruction and corroborate it after execution:

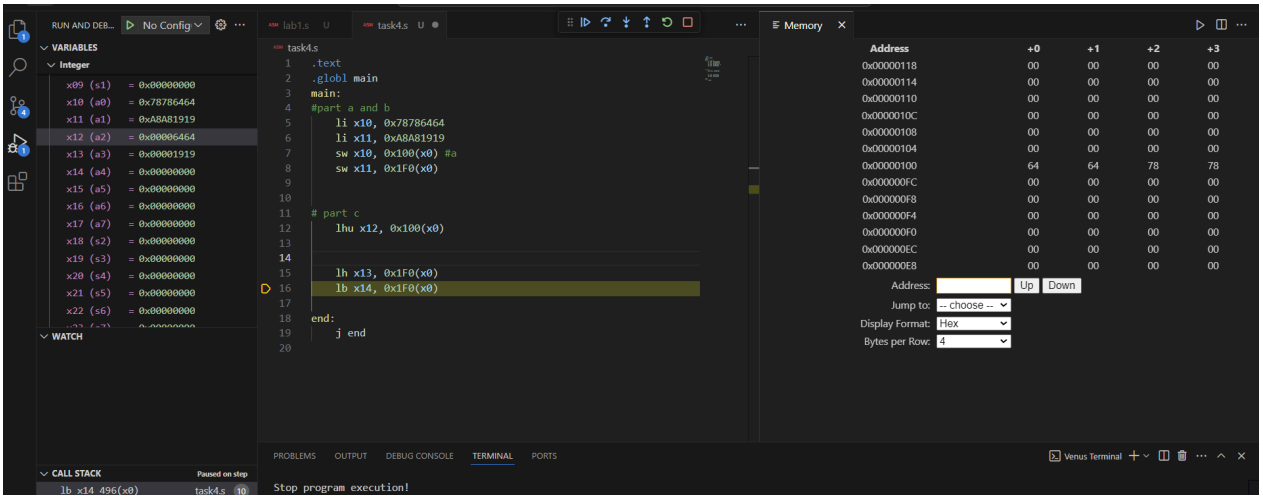
a) Store x10 as unsigned integer at address 0x100.



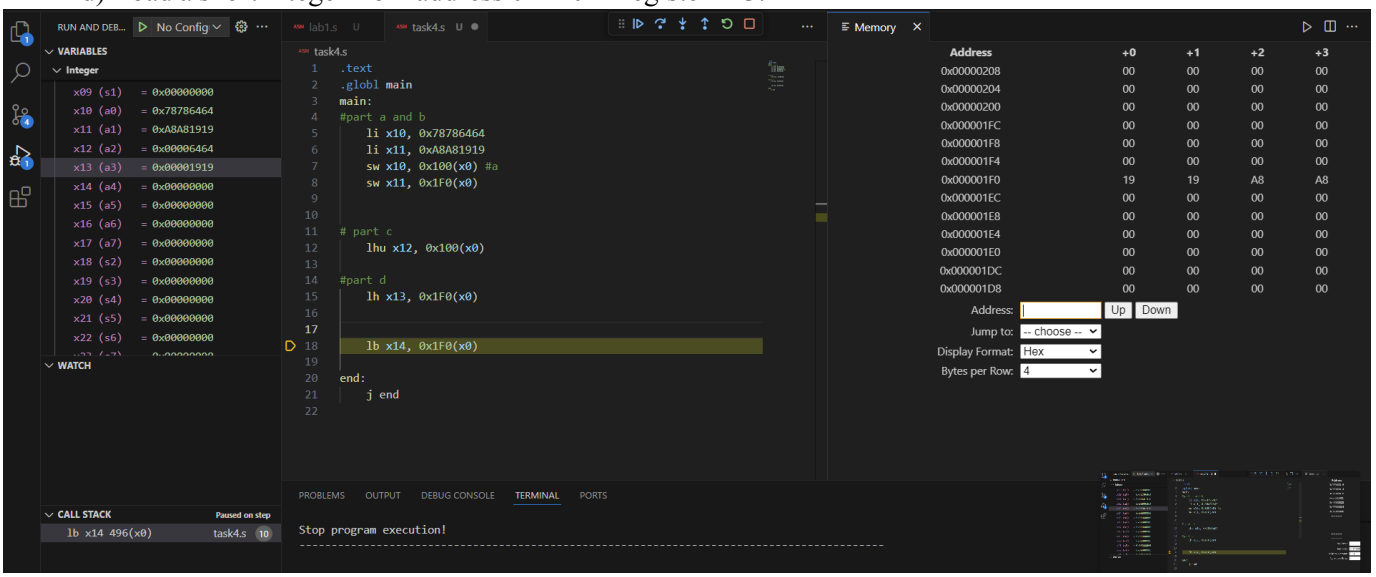
b) Store x11 as unsigned integer at address 0x1F0.



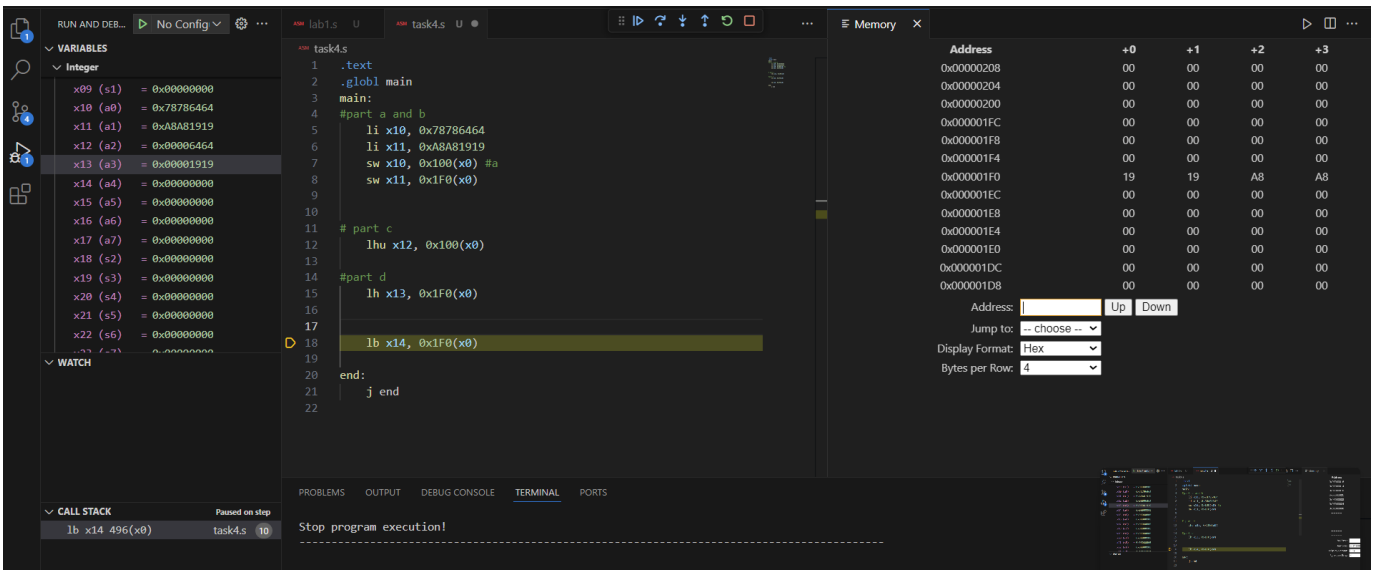
c) Load an unsigned short integer (two bytes) from address 0x100 in x12.



d) Load a short integer from address 0x1F0 in register x13.



e) Load a singed character from address 0x1F0 in register x14.



Task 4b – - Loop unrolling

Assume there are three character arrays a, b, and c located at addresses 0 x100, 0x200, 0 x300 respectively.

```
for (int i=0 ; i<4; i++ )
c [ i ]=a [ i ]+b [ i ] ; # c [ 0 ]=a [ 0 ]+b [0 ] ;
```

Write equivalent RISC-V code for the piece of code given. You have not studied loops yet, but the above code is manageable without loop instructions. Also assume that A is a character array, B is a short array , and C is an unsigned integer array.

Code

task4b.s

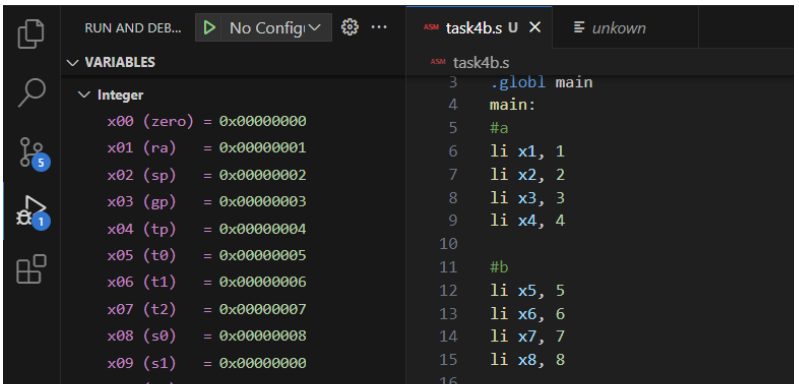
```
.globl main
main:
#a
li x1, 1
li x2, 2
li x3, 3
li x4, 4

#b
li x5, 5
li x6, 6
li x7, 7
li x8, 8

#saving a
sw x1, 0x100(x0)
sw x2, 0x101(x0)
sw x3, 0x102(x0)
sw x4, 0x103(x0)


#saving b
sw x5, 0x200(x0)
sw x6, 0x201(x0)
sw x7, 0x202(x0)
sw x8, 0x203(x0)
```

```
#loading a
lw x11, 0x100(x0)
lw x12, 0x101(x0)
lw x13, 0x102(x0)
lw x14, 0x103(x0)

#loading b
lw x15, 0x200(x0)
lw x16, 0x201(x0)
lw x17, 0x202(x0)
lw x18, 0x203(x0)
```

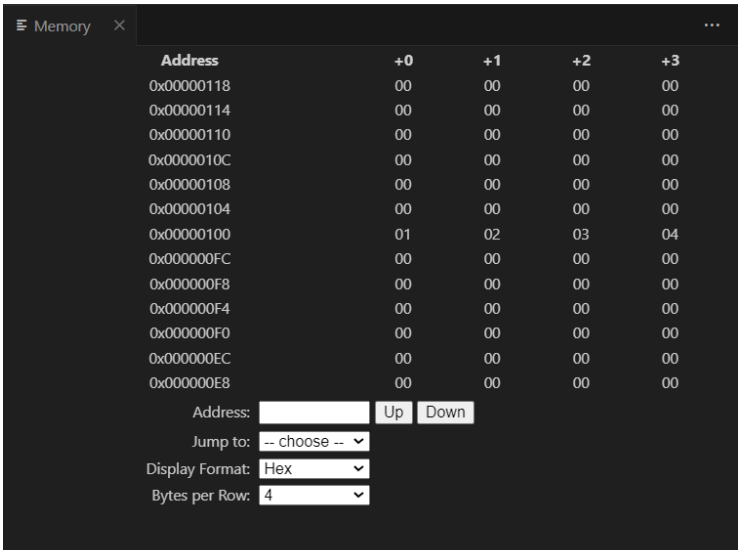
```
#adding both
add x21, x11, x15
add x22, x12, x16
add x22, x13, x17
add x24, x14, x18

#saving in c
sw x21, 0x300(x0)
sw x22, 0x301(x0)
sw x22, 0x302(x0)
sw x23, 0x303(x0)

end:
    j end
```

Output

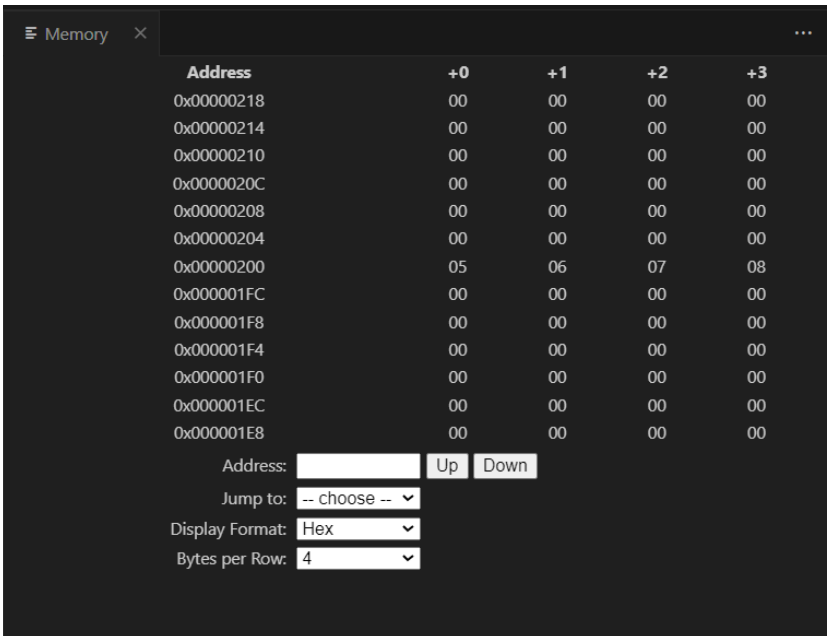
Array a and b loaded into registers



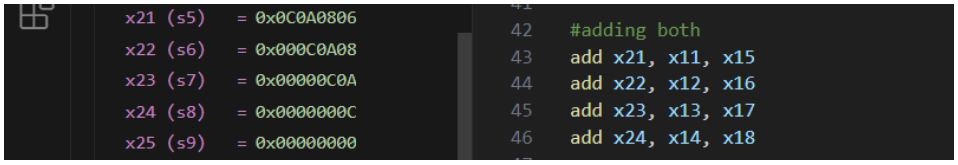
Saved a into memory (x100).



Saved b into memory (x200).

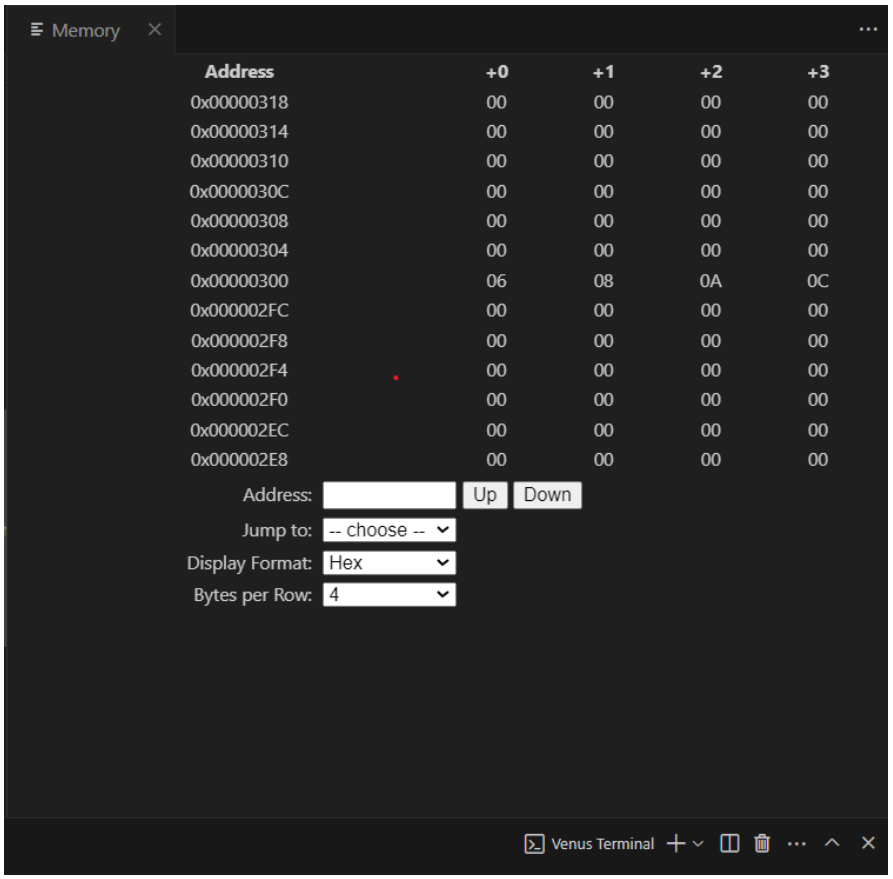


Result of a + b loaded in registers



Final Output

Result of a+ b stored in array c (x300)



Assessment Rubric

1: Getting Started with RISC-V (Assembly Language) in VS Code

Name:	Student ID:	section*:

Points Distribution

	Task No.	LR 2 Code	LR 5 Results
In - Lab	**Task 1**	/0	/15
	Task 2	/0	/15
	Task 3	/10	/5
	Task 4a	/10	/5
	Task 4b	/10	/10
Total Points: 100		**/30**	**/50**
CLO Mapped		CLO 2	

Affective Domain Rubric		Points	CLO Mapped
AR7	**Report Submission & Git Upload**	/10 & /10	CLO 2

CLO	Total Points	Points Obtained
2	100	
Total	**100**	

For description of different levels of the mapped rubrics, please refer to the Lab Evaluation Assessment Rubrics and Affective Domain Assessment Rubrics provided here.