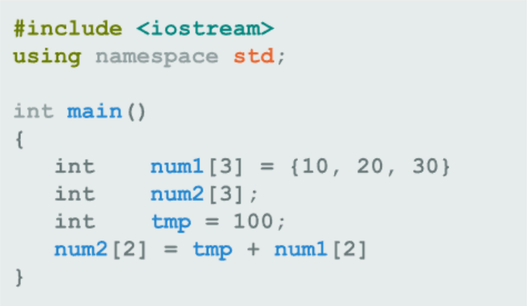
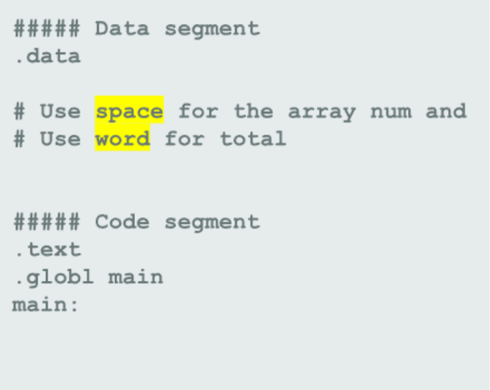
Assignment A P1-4 Assembly Program Lab 4. Access the elements in two Arrays

- Make an Assembly Program for the below C++ Program



```
#include <iostream>
using namespace std;

int main()
{
   int    num1[3] = {10, 20, 30}
   int    num2[3];
   int    tmp = 100;
   num2[2] = tmp + num1[2]
}
```



```
##### Data segment
.data

# Use space for the array num and
# Use word for total


##### Code segment
.text
.globl main
main:
```

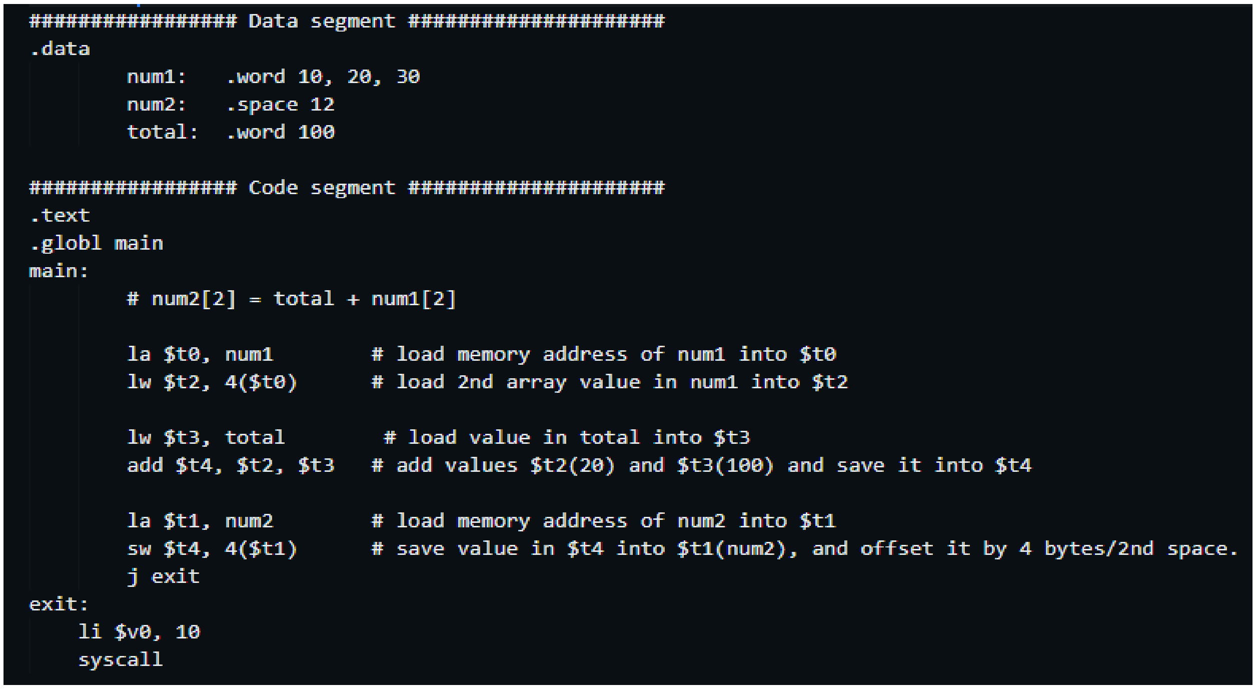
- Elaborate on your program
  - All register values
    - Elaborate the register values used in this program
  - The result value(Explain the values as decimal)
    - Which register has the result?

Submit Items:

GitHub Link to your file directly. (Please make it a link, not a text)

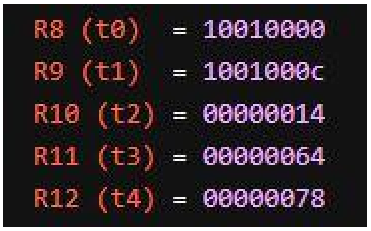
https://github.com/AVC-CS/p1-mtvonbargen/blob/739d2b29bb7085182686b26403daf50cf9f31b0b/P1A4.asm

Elaborate on your program

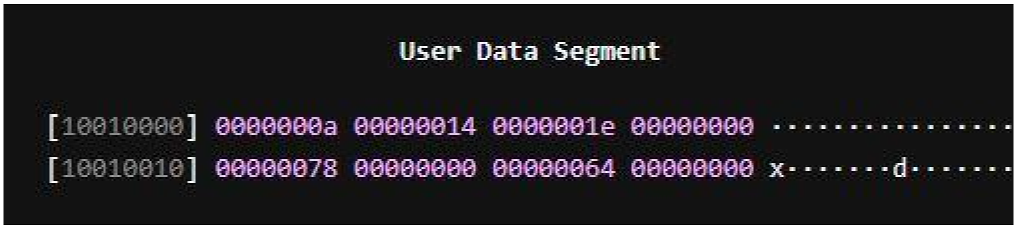


```
################# Data segment #####################
.data
        num1:   .word 10, 20, 30
        num2:   .space 12
        total:  .word 100

################# Code segment #####################
.text
.globl main
main:
        # num2[2] = total + num1[2]

        la $t0, num1        # load memory address of num1 into $t0
        lw $t2, 4($t0)      # load 2nd array value in num1 into $t2

        lw $t3, total        # load value in total into $t3
        add $t4, $t2, $t3   # add values $t2(20) and $t3(100) and save it into $t4

        la $t1, num2        # load memory address of num2 into $t1
        sw $t4, 4($t1)      # save value in $t4 into $t1(num2), and offset it by 4 bytes/2nd space.
        j exit
exit:
    li $v0, 10
    syscall
```

Program Results



```
R8 (t0)  = 10010000
R9 (t1)  = 1001000c
R10 (t2) = 00000014
R11 (t3) = 00000064
R12 (t4) = 00000078
```



```
                    User Data Segment

[10010000] 0000000a 00000014 0000001e 00000000 ················
[10010010] 00000078 00000000 00000064 00000000 x·······d·······
```

10010000 = 1st memory address starts here
1001000c = 2nd memory address starts here
00000014 = 20
00000064 = 100
00000078 = 120

Register Value. What is the purpose of this register? How is it updated throughout your program?

$t0 - Holds memory address of num1 and its arrays
$t1 - Holds memory address of num2 and its free array space
$t2 - Loaded the second integer from num1’s array into this register
$t3 - Loaded the integer from total
$t4 - Holds the sum of adding $t2 and $t3 together

The interaction between register and memory space

The interaction involves loading data from memory into registers for computation and storing results back into memory. Registers hold temporary data for operations, while memory provides large, slower storage for variables and arrays. Instructions like lw (load word) and sw (store word) facilitate this data transfer, with memory addresses accessed using base addresses and offsets. For example, data from num1[1] and total is loaded into registers, added together, and the result is stored back into num2[1], demonstrating the interaction between registers for computation and memory for storage.

Program logic

The program calculates the sum of num1[1] (20) and total (100), which equals 120, and stores the result into num2[1]. It uses registers to load data, perform the addition, and store the result back into memory.