**Bahria University, Lahore Campus**

Department of Computer Sciences

Lab Journal 011

**(Spring 2023)**

|  |  |  |
| --- | --- | --- |
| Course: | **Computer Architecture & Organization Lab** |  |
| Course Code: | CEL 221 | Max Marks: 40 |
| Faculty’s Name: | Maryam Munawar | Lab Engineer: |

Name: \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ Enroll No: \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

## Lab Tasks:

## Objective(s):

This is lab will introduce students to arrays, array input/output and array operations using MIPS

## Tool(s) used:

MARS 4.5

Since arrays can store LOTS of data, and since we have only a small (~32) number of registers, it is infeasible to use the registers for long-term storage of the array data.  Hence, arrays are stored in the Data Segment of a MIPS program.  Fundamentally, there are three operations which one can perform on an array:

* Getting the data from an array cell, e.g, x = list[i];
* Storing data into an array cell, e.g. list[i] = x;
* Determining the length of an array, i.e. list.length.

For purposes of this step in the lab, you may assume that the length of the array is 10.

To access the data in the array requires that we know the address of the data and then use the load word (lw) or store word (sw) instructions.  Words (which is how integers are stored) in MIPS take up 32 bits or 4 bytes. Therefore, if we have a declaration such as:

  list: .word 3, 0, 1, 2, 6, -2, 4, 7, 3, 7  
  
the address that is loaded by the instruction la $t3, list is the address of the first '3' in the list.  The address of the '0' is 4 greater than that number, and the address of the '6' is 16 greater than that number.

The following snippet of code will place the value of list[6] into the $t4:  
  
la $t3, list         # put address of list into $t3  
li $t2, 6            # put the index into $t2  
add $t2, $t2, $t2    # double the index  
add $t2, $t2, $t2    # double the index again (now 4x)  
add $t1, $t2, $t3    # combine the two components of the address  
lw $t4, 0($t1)       # get the value from the array cell

.

If we wish to assign to the contents of $t4 to list[6] instead, the last

line would simply be:

sw $t4, 0($t1)      # store the value into the array cell

## Tasks:

### Task1: 10 Minutes

Give answers to the following.

Declare an array of 5 int and assign them values

Assign a value of 3 to the 3rd element of the array declared above

Display the first value of the array

data

a:.asciiz "your first element of array is :"

myarray: .word 20

.text

addi $s0, $zero, 4

addi $s1, $zero, 10

addi $s2, $zero, 3

addi $s3, $zero, 4

addi $s4, $zero, 10

addi $t0, $zero, 0

sw $s0, myarray($t0)

addi $t0, $t0, 4

sw $s1, myarray($t0)

addi $t0, $t0, 4

sw $s2, myarray($t0)

lw $t6, myarray($zero)

li $v0, 4

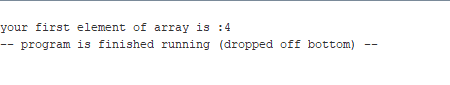
la $a0, a

syscall

li $v0, 1

addi $a0, $t6, 0

syscall



### Task2: 10 Minutes

.data

myarray: .space 100

newline: .asciiz "\n"

a: .asciiz "enter the element "

b: .asciiz "minimun value of this array is : "

.text

main:

addi $t0, $zero, 0

li $v0, 4

la $a0, a

syscall

li $v0 ,5

syscall

move $s0,$v0

sw $s0, myarray($t0)

addi $t0, $t0, 4

li $v0, 4

la $a0, a

syscall

li $v0 ,5

syscall

move $s1,$v0

sw $s1, myarray($t0)

addi $t0, $t0, 4

li $v0, 4

la $a0, a

syscall

li $v0 ,5

syscall

move $2,$v0

sw $s2, myarray($t0)

addi $t0, $t0, 4

li $v0, 4

la $a0, a

syscall

li $v0 ,5

syscall

move $s3,$v0

sw $s3, myarray($t0)

addi $t0, $t0, 4

li $v0, 4

la $a0, a

syscall

li ,$v0 ,5

syscall

move $s4,$v0

sw $s4, myarray($t0)

addi $t0, $t0, 4

li $v0, 4

la $a0, a

syscall

li ,$v0 ,5

syscall

move $s5,$v0

sw $s5, myarray($t0)

addi $t0, $t0, 4

li $v0, 4

la $a0, a

syscall

li ,$v0 ,5

syscall

move $s6,$v0

sw $s6, myarray($t0)

addi $t0, $t0, 4

li ,$v0 ,5

syscall

move $s7,$v0

sw $s7, myarray($t0)

addi $t0, $t0, 4

li $v0, 4

la $a0, a

syscall

li $v0 ,5

syscall

move $t3,$v0

sw $t3, myarray($t0)

addi $t0, $t0, 4

li $v0, 4

la $a0, a

syscall

li $v0 ,5

syscall

move $t1,$v0

sw $t1, myarray($t0)

li $v0, 4

la $a0, b

syscall

li $v0,1

move $a0,$s0

syscall

li $v0,10

syscall

Declare an array of 10 integers and get user input to fill the array values. Then, find the minimum value in the arrays.

.data

myarray: .space 100

newline: .asciiz "\n"

a: .asciiz "enter the element "

b: .asciiz "minimun value of this array is : "

.text

main:

addi $t0, $zero, 0

li $v0, 4

la $a0, a

syscall

li $v0 ,5

syscall

move $s0,$v0

sw $s0, myarray($t0)

addi $t0, $t0, 4

li $v0, 4

la $a0, a

syscall

li $v0 ,5

syscall

move $s1,$v0

sw $s1, myarray($t0)

addi $t0, $t0, 4

li $v0, 4

la $a0, a

syscall

li $v0 ,5

syscall

move $2,$v0

sw $s2, myarray($t0)

addi $t0, $t0, 4

li $v0, 4

la $a0, a

syscall

li $v0 ,5

syscall

move $s3,$v0

sw $s3, myarray($t0)

addi $t0, $t0, 4

li $v0, 4

la $a0, a

syscall

li ,$v0 ,5

syscall

move $s4,$v0

sw $s4, myarray($t0)

addi $t0, $t0, 4

li $v0, 4

la $a0, a

syscall

li ,$v0 ,5

syscall

move $s5,$v0

sw $s5, myarray($t0)

addi $t0, $t0, 4

li $v0, 4

la $a0, a

syscall

li ,$v0 ,5

syscall

move $s6,$v0

sw $s6, myarray($t0)

addi $t0, $t0, 4

li ,$v0 ,5

syscall

move $s7,$v0

sw $s7, myarray($t0)

addi $t0, $t0, 4

li $v0, 4

la $a0, a

syscall

li $v0 ,5

syscall

move $t3,$v0

sw $t3, myarray($t0)

addi $t0, $t0, 4

li $v0, 4

la $a0, a

syscall

li $v0 ,5

syscall

move $t1,$v0

sw $t1, myarray($t0)

li $v0, 4

la $a0, b

syscall

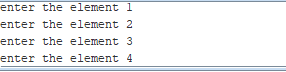
li $v0,1

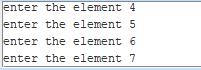
move $a0,$s0

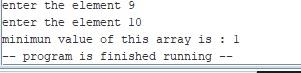
syscall

li $v0,10

syscall







### Task3: 10 Minutes

Write a program where you declare 3 arrays of 5 integers each. For two of the arrays get the values from the user. Add the respective elements of these two arrays and store the results in the third array.

Example:

Array1= 1 2 3 4 5

Array2 = 1 1 2 2 4

Array3 should be: 2 3 5 6 9

.data

a:.asciiz "your first element of array is :"

myarray1: .word 20

myarray2:.word 20

myarray:.word 20

.text

addi $s0, $zero, 1

addi $s1, $zero, 2

addi $s2, $zero, 3

addi $s3, $zero, 4

addi $s4, $zero, 5

addi $t0, $zero, 1

addi $t1, $zero, 1

addi $t2, $zero, 2

addi $t3, $zero, 2

addi $t4, $zero, 4

addi $t0, $zero, 0

sw $s0, myarray1($t0)

addi $t0, $t0, 4

sw $s1, myarray1($t0)

addi $t0, $t0, 4

sw $s2, myarray1($t0)

addi $t0, $t0, 4

sw $s3, myarray1($t0)

addi $t0, $t0, 4

sw $s4, myarray1($t0)

addi $t7, $zero, 0

sw $t1, myarray2($t7)

addi $t7, $t7, 4

sw $t2, myarray2($t7)

addi $t7, $t7, 4

sw $t3, myarray2($t7)

addi $t7, $t7, 4

sw $t4, myarray2($t7)

addi $t7, $t7, 4

sw $t5, myarray2($t7)

add $s0,$s0,$t1

add $s1,$s1,$t2

add $s2,$s2,$t3

add $s3,$s3,$t4

add $s4,$s4,$t5

addi $t6, $zero, 0

sw $s0, myarray($t6)

addi $t6, $t6, 4

sw $s1, myarray($t6)

addi $t6, $t6, 4

sw $s2, myarray($t6)

addi $t6, $t6, 4

sw $s3, myarray($t6)

addi $t6, $t6, 4

sw $s4, myarray($t6)

lw $s5, myarray($zero)

li $v0, 4

la $a0, a

syscall

li $v0, 1

addi $a0, $s5, 0

syscall

li $v0,10

syscall



.data

a:.asciiz "your first element of array is :"

myarray1: .word 20

.text

### Task4: 10 Minutes

Write a program where you declare an array and pass it to a function. The function should count the number of even numbers in the array and return it.

.data

a:.asciiz "your first element of array is :"

myarray1: .word 20

.text

addi $s0, $zero, 1

addi $s1, $zero, 2

addi $s2, $zero, 3

addi $s3, $zero, 4

addi $s4, $zero, 5

addi $t0, $zero, 0

sw $s0, myarray1($t0)

addi $t0, $t0, 4

sw $s1, myarray1($t0)

addi $t0, $t0, 4

sw $s2, myarray1($t0)

addi $t0, $t0, 4

sw $s3, myarray1($t0)

addi $t0, $t0, 4

sw $s4, myarray1($t0)

func:

div $t1,$s0,2

mfhi $t1

beq $t1,0,print0

div $t2,$s1,2

mfhi $t2

beq $t2,0,print1

div $t3,$s2,2

mfhi $t2

beq $t3,0,print2

div $t4,$s3,2

mfhi $t4

beq $t4,0,print3

div $t5,$s4,2

mfhi $t5

beq $t5,0,print4

print0 :

li $v0,1

move $a0,$s0

syscall

print1 :

li $v0,1

move $a0,$s1

syscall

print2 :

li $v0,1

move $a0,$s2

syscall

print3 :

li $v0,1

move $a0,$s3

syscall

print4 :

li $v0,1

move $a0,$s4

syscall

**Lab Grading Sheet :**

|  |  |  |  |
| --- | --- | --- | --- |
| **Task** | **Max Marks** | **Obtained Marks** | **Comments(*if any*)** |
| a. | 10 |  |  |
| b. | 10 |  |  |
| c. | 10 |  |  |
| d | 10 |  |  |
| **Total** | **40** |  | **Signature** |

**Note : Attempt all tasks and get them checked by your Instructor**