Preparation [Total max. mark: 1]  
The main goal of today’s lab is to become familiar with manipulating arrays using pointers and indices, and to reinforce the high level language idioms introduces in the lectures. References: the textbook Ed2, paragraphs 3.5, 3.11 or Ed3:  
2.6, 2.15, Ed4: 2.7, 2.14, Ed5: 2.7, 2.14, and the lecture notes.  
  
 **1. Describe below in writing (or in tabular format) INDICES and POINTERS methods of ARRAY traversal by comparing their significant aspects. [**0.5 mark**].**

The loop structure that is used is a little different from each other; we access the values in two different ways; either:

1. Indexing

Address Calculation: Using the index\*4 + Array-Address[first base]

In this case, we are using the index as means to discontinue the traversing; by comparing it to the size of the array.

1. Pointers

Address Calculation: Using the base address and then continuously doing the following calculation 4+Address[i]. In

In this case, we are using the size\*4+array[0] as the comparison value to discontinue the traversing; if the current address+4 is > than the size\*4+address then the traversing will discontinue.

**2. Illustrate your explanation on array traversal using INDICES and POINTERS methods with a hand drawing and a few lines of code [0.5 mark].**

**Method for Accessing Information for Array Data Structures**

An array can be accessed and traversed using an indexing or pointer method.

1. Indexing Method:

The value of the index is multiplied by 4 and then added to the base address of the array; and then accessed (stored or loaded) afterwards.

addi $t0, $t0, 0 # I = 0

**loop:**

add $t1, $t0, $t0

add $t1, $t1, $t1 #i\*4

add $t2, $a0, $t1 #i\*4 + Arraybase

lw $t3, 0($t2)

1. Pointer Version

The base address of the array is added by four (or some other number depending on the data structure requirement) progressively, and then the data can be accessed, manipulated or organized by using this base address and the corresponding operations (or otherwise).

Addi $t0, $a0, $0 #base address

add $t1, $a1, $a1

add $t1, $t1, $t1 #size\*4

add $t2, $a0, $t1

**loop:** lw $t3, 0($t0)

addi $t0, $t0, 4 # p = p+4

slt $t4, $t0, $t2

bne $t3, $zero, loop