CSC 256 - Spring 2016
Pointer vs. Array Access Translation

Array Access (char)	Pointer Access (char)

`char str[6];`
`for(i=0;i<6;i++)`
`  str[i] = 0xa;`
	`char str[6];`
	`char *ptr = str;`
	`for(i=0;i<6;i++){`
	`  *ptr = 0xa;`
	`  ptr++;`
	`}`

`  # $s0 -> i`
`  # $s1 -> base`
`  # $t0 -> 6`
`  # $t1 -> 0xa`
`  # $t2 -> base+offset`
`  .data`
`str:  .byte 0:6`
`  .text`
`  li  $t0,6`
`  li  $t1, 0xa`
`  li  $s0,0`
`  la  $s1, str`
`loop: add  $t2,  $s1, $s0`
`  sb  $t1, ($t2)`
`  addi  $s0, $s0, 1`
`  blt  $s0, $t0, loop`
	`  # $s0 -> i`
	`  # $s1 -> ptr`
	`  # $t0 -> 0xa`
	`  # $t1 -> 6`
	`  #`
	`  .data`
	`str:  .byte 0:6`
	`  .text`
	`  li  $t0, 0xa`
	`  li  $t1, 6`
	`  li  $s0,0`
	`  la  $s1, str`
	`loop: sb  $t0, ($s1)`
	`  addi  $s1, $s1, 1`
	`  addi  $s0, $s0, 1`
	`  blt  $s0, $t0, loop`

Array Access (Int)	Pointer Access (Int)

`int x[6];`
`for(i=0;i<6;i++)`
`  x[i] = i;`
	`int x[6];`
	`int *ptr = x;`
	`for(i=0;i<6;i++){`
	`  *ptr = i;`
	`  ptr++;`
	`}`

`  # $s0 -> i`
`  # $s1 -> base`
`  # $t0 -> 6`
`  # $t1 -> base+offset`
`  .data`
`x:  .word 0:6`
`  .text`
`  li  $t0,6`
`  li  $s0,0`
`  la  $s1, x`
`loop: sll  $t1, $t1, 2`
`  add  $t1, $t1, $s1`
`  sw  $s0, ($t1)`
`  addi  $s0, $s0, 1`
`  blt  $s0, $t0, loop`
	`  # $s0 -> i`
	`  # $s1 -> ptr`
	`  # $t0 -> 6`
	`  #`
	`  .data`
	`x:  .word 0:6`
	`  .text`
	`  li  $t0, 6`
	`  li  $s0,0`
	`  la  $s1, x`
	`loop: sb  $s0, ($s1)`
	`  addi  $s1, $s1, 4`
	`  addi  $s0, $s0, 1`
	`  blt  $s0, $t0, loop`