

CS 2734, Arrays and Functions, Page 1 of 1

```
# CS 2734, Computer Organization II, Fall 2001
# MIPS program showing arrays and functions
# #include <stdio.h>
# void Write_array(int [], int);
# void main(void)
# {
#     int A[10] = {10, 20, 30, 40, 50, 60,
#                 70, 80, 90, 100};
#     int i;
#     A[3] = 47;
#     i = A[5];
#     printf("Value in i: %i\n", i);
#     Write_array(A, 10);
# }
# void Write_array(int A[10], int n)
# {
#     int j;
#     printf("Values in array A: ");
#     for (j = 0; j < n; j++)
#         printf("%i ", A[j]);
#     printf("\n");
# }
# ten42% write_array
# Value in i: 60
# Values in array A: 10 20 30 47 50 60 70 80 90 100
#####
.globl main
main:    addu    $s7, $zero, $ra
        .data
A:       .word   10, 20, 30, 40, 50, 60, 70, 80, 90, 100
        .text
        la      $t0, A
        addi    $t2, $zero, 47
        sw      $t2, 12($t0)
        lw      $t1, 20($t0)
## Print value in $t1
        li      $v0, 4
        la      $a0, Regt1
        syscall
        li      $v0, 1
        move    $a0, $t1
        syscall
        jal     Newl
## Print the array
        li      $v0, 4
        la      $a0, Amess
        syscall
        la      $a0, A
        li      $a1, 10
        jal     Write_array
        jal     Newl
##### Finish main#####
        addu    $ra, $zero, $s7
        jr      $ra
##### End of main function #####
```

```
##### function definitions #####
##### write an array #####
Write_array:
        addi    $sp, $sp, -4      # room for $ra on stack
        sw      $ra, 0($sp)      # save $ra because not leaf
        ## initialization for loop
        move    $s0, $a0         # $s0 = $a0 = start of A
        move    $s1, $a1         # $s1 = $a1 = N
        move    $t1, $zero       # start $t1 = 0, the index
LoopA:   beq     $s1, $t1, EndA    # if (N == index) goto EndA
        ## write value for A[i]
        addu    $t2, $t1, $t1
        addu    $t2, $t2, $t2
        addu    $t2, $s0, $t2    # $t2 = index*4 + start of A
        li      $v0, 1
        lw      $a0, 0($t2)      # integer to print
        syscall
        ## write a blank
        jal     Blan
        addi    $t1, $t1, 1
        j       LoopA
EndA:    lw      $ra, 0($sp)
        addi    $sp, $sp, 4
        jr      $ra

##### write newline #####
Newl:    li      $v0, 4
        la      $a0, Newline
        syscall
        jr      $ra

##### write blank #####
Blan:    li      $v0, 4
        la      $a0, Blank
        syscall
        jr      $ra

        .data
Blank:    .ascii  " "
Newline:  .ascii  "\n"
Regt1:    .ascii  "Value in Register $t1: "
Amess:    .ascii  "Values in array A: "
Thatsall: .ascii  "Th-th-th-thats all folks!\n"
```

```
##### output #####
# four06% spim -file write_array.s
# Value in Register $t1: 60
# Values in array A: 10 20 30 47 50 60 70 80 90 100
#####
```