```
# CS 2734, Fall 2001
# Array loop examples, using index and using pointer
# Print 10 integers
#  address of A and later B is in $s5

        .globl main
main:                           # main has to be a global label
        addu    $s7, $0, $ra    # save return address in global register

        la      $s5, A          # $s5 has the starting address of A
## Start of simple loop example
        addi    $s3, $zero, 0   # s3 = 0, used for index i
        addi    $s4, $zero, 10  # s4 = 10


Loop:
        add     $t1, $s3, $s3   # Temporary register $t1 = 2*i
        add     $t1, $t1, $t1   # Double again so temporary register has 4*i
        add     $t1, $t1, $s5   # Add A start address so t1 has addr of A[i]
        lw      $t0, 0($t1)     # Temporary register $t0 has value of A[i]
## Print array element and a blank
        li      $v0, 1          # print_int (system call 1)
        add     $a0, $0, $t0    # put value A[i] == $t0 in $a0 for printing
        syscall
        li      $v0, 4          # print_str (system call 4)
        la      $a0, blank      # takes the address of string as an argument
        syscall
## Ready for next loop
        addi    $s3, $s3, 1
        bne     $s4, $s3, Loop  # if (i != 10) go to Loop

## Print a newline
        li      $v0, 4          # print_str (system call 4)
        la      $a0, newl       # takes the address of string as an argument
        syscall



###############################################################################
## Start of second simple loop example
        la      $s5, B
        addi    $s3, $zero, 0   # s3 = 0, used for index i
        addi    $s4, $zero, 10  # s4 = 10
        add     $s6, $zero, $s5 # used for pointer into B, address of B[i]

Loop2:
        lw      $t0, 0($s6)     # Temporary register $t0 has value of B[i]
## Print array element and a blank
        li      $v0, 1          # print_int (system call 1)
        add     $a0, $0, $t0    # put value B[i] == $t0 in $a0 for printing
        syscall
        li      $v0, 4          # print_str (system call 4)
        la      $a0, blank      # takes the address of string as an argument
        syscall
## Ready for next loop
        addi    $s3, $s3, 1
        addi    $s6, $s6, 4
        bne     $s4, $s3, Loop2 # if (i != 10) go to Loop2

## Print a newline
        li      $v0, 4          # print_str (system call 4)
        la      $a0, newl       # takes the address of string as an argument
        syscall

## Usual stuff at the end of the main
        addu    $ra, $0, $s7    # restore the return address
        jr      $ra             # return to the main program

        .data
        .align  2               # Let's make sure that it's aligned
A:      .word   11,13,15,17,19,21,23,25,27,29
B:      .word   15,20,25,30,35,40,45,50,55,60
blank:  .asciiz " "  # string to print
newl:   .asciiz "\n" # string to print
### output
# four06% spim -file array.s
# SPIM Version 6.0 of July 21, 1997
# Copyright 1990-1997 by James R. Larus (larus@cs.wisc.edu).
# All Rights Reserved.
# See the file README for a full copyright notice.
# Loaded: /usr/local/lib/trap.handler
# 11 13 15 17 19 21 23 25 27 29
# 15 20 25 30 35 40 45 50 55 60
```