```
#             Lecture 15 Demo Program 3, Nested Loops
# SPIM code to initialize the elements of a 2-d array using a pointer

#                Register Assignments

# $t0 = Pointer to current beginning of row
# $t1 = Row counter and index
# $t2 = Column counter and index
# $t3 = Pointer to current address to store data
# $t4 = Value to be stored in each array element

       .data
ar2:   .space 1600         # Array of 400 integers (4 bytes each) in 20X20 array

        .text
main:   la $t0,ar2         # Initialize pointer to start of array
        move $t1,$0        # Initialize row counter/index
        li $t4,0x4ff6      # Put value to be loaded in array in $t4

rloop:  move $t2,$0        # Initialize column counter/index
        move $t3,$t0       # Initialize col. pointer to 1st element of row

cloop:  sw $t4,0($t3)      # Store value in current array element
        addi $t2,$t2,1     # Increment column counter/index by 1
        beq $t2,20,nxtrow  # Go to next row if column counter = 20
        addi $t3,$t3,4     # Increment the column pointer
        j cloop            # Go back and do another column

nxtrow: addi $t1,$t1,1     # Increment row counter/index by 1
        beq $t1,20,end     # Leave row loop if row counter = 20
        add $t0,$t0,80     # Increment the beginning-of-row pointer by
                           #    the number of bytes in a row
        j rloop            # Start next row

end:    li $v0,10          # Reach here if row loop is done
        syscall            # End of program
#
#       End of file Lecture 15 Demo Program 3
```