# Author : Kevin De La Torre ( 010806250 )

# Class : CS 264

# Professor: Salloum

# Project : Lab 1

.data # Initialize variable declaration block

intArray: .space 80 # Allocate 80 bytes ( 4 per int )

initPrompt: .asciiz "\nEnter 20 positive integers\n"

inPrompt: .asciiz ">> "

nPrompt: .asciiz "\nEnter number of integers per line ( <= 20 ): "

invPrompt: .asciiz "Invalid entry"

newLine: .asciiz "\n"

space: .asciiz " "

.globl main # Declare starting point

.text # Initialize instruction block

# Program accepts 20 integers from user then prints them out in different formats

main: la $t0, intArray # load intArray into $t0

li $t1, 20 # Counter variable

la $a0, initPrompt

li $v0, 4 # Print string

syscall

# Initial input loop

inLoop: la $a0, inPrompt

li $v0, 4

syscall

li $v0, 5 # Read int

syscall

sw $v0, 0( $t0 ) # Return is stored in $v0, then we store at current head of $t0

addi $t0, 4 # Move head of $t0 over 4, which means move onto next int in array

addi $t1, -1 # Decrement our counter

bgtz $t1, inLoop # If $t0 greater than zero -> loop

addi $t0, -80 # After getting our integers, move the $t0 back to root

li $t1, 20 # Reset counter to 20

# First format - One per line

loop1: lw $a0, 0( $t0 )

li $v0, 1 # Print int

syscall

la $a0, newLine

li $v0, 4

syscall

addi $t0, 4

addi $t1, -1

bgtz $t1, loop1

# Prepare for next loop

addi $t0, -80 # Have to move back again because reading from beginning

li $t1, 20

la $a0, newLine

li $v0, 4

syscall

# Second format - Single line, sep by spaces

loop2: lw $a0, 0( $t0 )

li $v0, 1

syscall

la $a0, space

li $v0, 4

syscall

addi $t0, 4

addi $t1, -1

bgtz $t1, loop2

addi $t0, -4 # Go to before last index instead of after

li $t1, 20

la $a0, newLine

li $v0, 4

syscall

# Third format - Reverse single line, sep by spaces

loop3: lw $a0, 0( $t0 )

li $v0, 1

syscall

la $a0, space

li $v0, 4

syscall

addi $t0, -4 # Go back one integer in array

addi $t1, -1

bgtz $t1, loop3

# Prepwork for final loop

addi $t0, 4 # We subtracted one int too many at the end of last loop, out of array

li $t1, 20

la $a0, newLine

li $v0, 4

syscall

j nInput # jump past errRange label

# If out of range, display prompt, then go back

errRange: la $a0, invPrompt

li $v0, 4

syscall

j nInput

# Get n from user

nInput: la $a0, nPrompt

li $v0, 4

syscall

li $v0, 5

syscall

blez $v0, errRange # If n less than 0, give error

bgt $v0, $t1, errRange # If greater than 20, give error

move $t2, $v0 # $t2 = n

move $t3, $t2 # Create N counter for following loop

# Fourth format - Print n integers per line

loop4: lw $a0, 0( $t0 )

li $v0, 1

syscall

la $a0, space

li $v0, 4

syscall

addi $t0, 4 # Move array

addi $t1, -1 # Decrement loop counter

addi $t3, -1 # Decrement n counter

beqz $t1, exit # If out of numbers in array exit

bgtz $t3, loop4 # If haven't reached n print next num

la $a0, newLine # Go to next line

li $v0, 4

syscall

move $t3, $t2 # reset counter

j loop4

exit: li $v0, 10 # Exit program

syscall