

```

# Lecture 17, Demo Program 1: Character List Alphabetization
# Alphabetizes a sequence of ASCII lower-case letters and prints out the resulting
# alphabetized list.
# The list of letters is in the data declaration and contains all 26 alphabet
# letters in random order.

# $t0 -- Pointer to current spot in letters
# $t1 -- Holds the "upstream compare character"
# $t2 -- Holds the current character being analyzed
# $t7 -- Pointer to the first character in string

.main: .text
main:  la $t0,string    # Load the string address into $t0
       la $t7,string    # Load the string address into $t7
comp:   lb $t1,0($t0)    # Load first two characters to be compared
       lb $t2,1($t0)
       beqz $t2,done    # If the new character = 0, done
       ble $t1,$t2,count # If characters in correct order, get next character
       jal rev          # Characters not in correct order; go to reverse
       j comp           # Character in correct position; get next character
count:  addi $t0,$t0,1    # Increment current character address
       j comp           # Return to next character compare

done:   la $a0,string    # Print out alphabetized string + CR
       li $v0,4
       syscall
       li $v0,10        # Done; end program.
       syscall

# Character reverse routine follows
rev:    sub $sp,$sp,4    # Store contents of $ra on the stack
       sw $ra,($sp)      # Decrement stack pointer.
       sb $t1,1($t0)    # Exchange two character positions
       sb $t2,0($t0)
       beq $t0,$t7,goback # If at first position in the string, done

       sub $t0,$t0,1    # Decrement the letter pointer.
       lb $t1,0($t0)    # Compare letter to next "upstream" letter
       lb $t2,1($t0)
       ble $t1,$t2,goback # If letter is properly placed, done
       jal rev          # Not done yet; move back another position

goback: addi $t0,$t0,1    # Reverse done; move back to current position
        lw $ra,($sp)
        addi $sp,$sp,4
        jr $ra

.data
string: .ascii "qwertyuiopasdfghjklzxcvbnm"    # Characters list
# End of Lecture 17 Demo Program 1.

```