/\*

Write a program that accepts a string from the user and a character from the user and does the following operations:

1. computes the length of the string.

2. converts the string to lowercase.

3. count the occurence of the character in the string.

4. remove the locations of the character in the string.

Example Output:

Please enter in a string of a maximum of 30 letters: iloVeAssEmBlyLanguAge

Please enter a letter: e

The string in lower case is: iloveassemblylanguage

The length of the string is 21 characters

The letter e appears 3 times

The revised string is: ilovassmblylanguag

Would you like to enter in another string (y/n)? n

Good bye.

\*/

!Variables and Macros

define(str, g1)

define(len, l4)

.section ".data"

input: .asciz " "

nl: .byte 0 !for newline

format: .asciz "%s"

nl2: .byte 0

charinput: .byte 0

charformat: .byte " %c"

stringMessage: .asciz "Please enter in a string of a maximum of 30 letters: "

letterMessage: .asciz "Please enter a letter: "

lengthMessage: .asciz "The length of the string is %d characters \n"

lowercaseMessage: .asciz "The string in lowercase is %s \n"

!testing

test: .asciz "String is: %s"

testchar: .asciz "Char is: %c \n"

.align 4

.section ".text"

.global main

main:

save %sp, -96, %sp

set stringMessage, %o0

call printf

nop

set format, %o0

set input, %o1

set nl, %o2

call scanf

nop

mov %o1, %str

ldub [%o2], %l0

ld [%str], %l7

/\* Uncomment to see string value.

set test, %o0

call printf

nop

\*/

clr %o1

clr %o2

clr %o0

clr %len

/\* Code for getting character

set letterMessage, %o0

call printf

nop

set charformat, %o0

set charinput, %o1

call scanf

nop

ldub [%o1], %l1

set testchar,%o0

call printf

nop

\*/

!length and lowercase

lengthloop:

ldub [%str], %l5 !load byte by byte from the string

cmp %l5, 0 !compare will null terminator

be donelength !branch if you found the null terminator

inc %str

inc %len

ba lengthloop

nop

donelength: !print the length of the string

mov %len, %o1

set lengthMessage, %o0

call printf

nop

/\* lowercase loop \*/

lowercaseloop:

ldub [%l7], %l6

cmp %l6, 0

be donelowercase

nop

cmp %l6, 'a' !97 in ASCII = 0x61

bl donelowercase

nop

cmp %l6, 'z' !122 in ASCII = 0x7a

bg donelowercase

nop

sub %l6, 0x20, %l6 !0x20 is 6th bit pivot in ASCII

stb %l6, [%l7]

donelowercase:

inc %l7

ba lowercaseloop

nop

done:

set lowercaseMessage, %o0

ld [%l7], %o1

call printf

nop

ret

restore