; \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

; File: COMPORGHW3.S

; Programmer: Manoj Vasa

; Description: Prints original string, analyzes each

; character for vowel, capitalizes lower case vowels, and

; prints altered string.

; Date: Feb. 25, 2016

; \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

AREA Homework3, CODE

SWI\_swrite EQU &0 ;output character in r0

SWI\_exit EQU &11 ;finish program

ENTRY

Main

ADR r1, String ;store string address in r1

ADR r2, String ;store string address in r2

MOV r3, #0 ;initialize counter

OriginalPrint LDRB r0, [r1], #1 ;load next byte

CMP r0, #0 ;check for null terminator of string

SWINE SWI\_swrite ;if not equal, then print

BNE OriginalPrint ;loop back to OriginalPrint if not equal to 0

VowelCheck LDRB r0, [r2], #1 ;load next byte of string

CMP r0, #0 ;check for null terminator of string

BNE CheckLower ;if not equal to null terminator go to CheckLower

ADR r1, StringNum ;store string in r1

NumLoop LDRB r0, [r1], #1 ;load next byte

CMP r0, #0 ;make sure if its null terminator

SWINE SWI\_swrite ;if not null terminator, then print

BNE NumLoop ;if not null terminator, then continue to NumLoop

;MOV r0, r3 ;store counter in r0

;LDRB r0, [r0] ;load counter

;SWI SWI\_swrite ;print counter

;ADR r1, StringNum

;MOV r4, #8 ;count nibbles

;ParseLoop MOV r0,r1,LSR #28 ;reach top nibble

;CMP r0, #9 ;compare nibble with 9

;ADDGT r0, r0, #"A"-10 ;if greater than, add 55 to value in r0 and store in r0

;ADDLE r0, r0, #"0" ;if less than, add 48 to value in r0 and store in r0

;SWI SWI\_swrite ;print character

;MOV r1, r1, LSL # 4 ;shift left 4 bits

;SUBS r2, r2, #1 ;decrement nibble counter

;BNE ParseLoop ;go to ParseLoop if there are more nibbles

;MOV pc, r14 ;return

SWI SWI\_exit ;exit the program

CheckLower CMP r0, #96 ;compare byte to 96 to check for lower case vowel

BGE LowerVowela ;if greater than or equal to 96, go to LowerVowela

TEQ r0, #65 ;compare byte with decimal value of upper case A

BEQ PrintNormal ;if equal go to PrintNormal

TEQ r0, #69 ;compare byte with decimal value of upper case E

BEQ PrintNormal ;if equal go to PrintNormal

TEQ r0, #73 ;compare byte with decimal value of upper case I

BEQ PrintNormal ;if equal go to PrintNormal

TEQ r0, #79 ;compare byte with decimal value of upper case O

BEQ PrintNormal ;if equal go to PrintNormal

TEQ r0, #85 ;compare byte with decimal value of upper case U

BEQ PrintNormal ;if equal go to PrintNormal

PrintNormal SWIEQ SWI\_swrite ;if equal print character

ADDEQ r3, r3, #1 ;if equal increment counter

BNE WriteOthers ;if not equal, go to WriteOthers

B VowelCheck ;unconditional jump back to VowelCheck

LowerVowela TEQ r0, #97 ;compare byte with decimal value of lower case a

BEQ PrintUpper ;if equal go to printUpper

TEQ r0, #101 ;compare byte with decimal value of lower case e

BEQ PrintUpper ;if equal go to printUpper

TEQ r0, #105 ;compare byte with decimal value of lower case i

BEQ PrintUpper ;if equal go to printUpper

TEQ r0, #111 ;compare byte with decimal value of lower case o

BEQ PrintUpper ;if equal go to printUpper

TEQ r0, #117 ;compare byte with decimal value of lower case u

PrintUpper SUBEQ r0, r0, #32 ;assign lower case letters upper case letter values

SWIEQ SWI\_swrite ;print character

ADDEQ r3, r3, #1 ;increment counter

BNE WriteOthers ;if not equal, go to WriteOthers

B VowelCheck ;unconditinonal jump back to VowelCheck

WriteOthers SWI SWI\_swrite ;print character

B VowelCheck ;unconditinonal jump back to VowelCheck

String DCB "aEIOU all the vowels are 23456 amazing. numbers 89 78 0654", &0a, 0

StringNum DCB "16", &0a, 0

ALIGN

END