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

; File: COMPORGHW6.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 Homework6, CODE

SWI\_write EQU &0 ;output character in r0

SWI\_exit EQU &11 ;finish program

ENTRY

Main

ADR r13, Stack

ADR r1, firstStr

MOV r2, #0

ADR r3, Key

LDR r3, [r3]

B JumpTable

Increment ADD r2, r2, #1

JumpTable ADR r4, SubTable

LDR pc, [r4, r2, LSL#2]

SubTable DCD Encryption

DCD Decryption

DCD Compare

DCD Exit

Encryption MOV r5, r1

prntLoop LDRB r0, [r5], #1

CMP r0, #0

SWINE SWI\_write

BNE prntLoop

MOV r5, r1

nextletter LDRB r0, [r5], #1

CMP r0, #0

BNE Encrypt

B Increment

Encrypt MOV r6, r0

AND r6, r6, #0XFF000000 ;first byte masked--have to move to third byte------right shift 16

MOV r7, r0

AND r7, r7, #0X00FF0000 ;second byte masked-have to move to fourth byte-----right shift 16

MOV r8, r0

AND r8, r8, #0X0000FF00 ;third byte masked--have to move to first byte-------left shift 16

MOV r9, r0

AND r9, r9, #0X000000FF ;fourth byte masked-have to move to second byte------left shift 16

MOV r6, r6, LSR#16

MOV r7, r7, LSR#16

MOV r8, r8, LSL#16

MOV r9, r9, LSL#16

MOV r10, #0

ORR r10, r10, r6

ORR r10, r10, r7

ORR r10, r10, r8

ORR r10, r10, r9

EOR r10, r10, r3

;store r10 in array and duplicate array before printing

MOV r11, r10

Print MOV r7,#8 ;count of nibbles = 8

LOOP MOV r0,r11,LSR #28 ;get top nibble

CMP r0, #9 ;hexanumber 0-9 or A-F

ADDGT r0,r0, #"A"-10 ;ASCII alphabetic

ADDLE r0,r0, #"0" ;ASCII numeric

SWI SWI\_write ; print character

MOV r11,r11,LSL #4 ;shift left one nibble

SUBS r7,r7, #1 ;decrement nibble count

BNE LOOP ;if more nibbles,loop back

EOR r10, r10, r03

MOV r6, r10

AND r6, r6, #0XFF000000 ;first byte masked--have to move to third byte------right shift 16

MOV r7, r10

AND r7, r7, #0X00FF0000 ;second byte masked-have to move to fourth byte-----right shift 16

MOV r8, r10

AND r8, r8, #0X0000FF00 ;third byte masked--have to move to first byte-------left shift 16

MOV r9, r10

AND r9, r9, #0X000000FF ;fourth byte masked-have to move to second byte------left shift 16

MOV r6, r6, LSR#16

MOV r7, r7, LSR#16

MOV r8, r8, LSL#16

MOV r9, r9, LSL#16

MOV r10, #0

ORR r10, r10, r6

ORR r10, r10, r7

ORR r10, r10, r8

ORR r10, r10, r9

MOV r0, r10

SWI SWI\_write

B nextletter

Decryption

;printLoop ;LDRB r0, [r6], #1

;CMP r0, #0

;SWINE SWI\_write

;BNE printLoop

B Increment

Compare

;printLoop1 ;LDRB r0, [r7], #1

;CMP r0, #0

;SWINE SWI\_write

;BNE printLoop1

B Increment

Exit SWI SWI\_exit

firstStr DCB "first string.", 0

Space DCB " "

Key DCD 0xAAEE3425

;firstByte DCD 0XFF000000

;secondByte DCD 0X00FF0000

;thirdByte DCD 0X0000FF00

;fourthByte DCD 0X000000FF

Stack % 128\*4

END