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

; 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 r3, firstStr

MOV r1, #0

ADR r4, Key

LDR r4, [r4]

B JumpTable

Increment ADD r1, r1, #1

JumpTable ADR r2, SubTable

LDR pc, [r2, r1, LSL#2]

SubTable DCD Encryption

DCD Decryption

DCD Compare

DCD Exit

Encryption MOV r8, r3

prntLoop LDRB r0, [r8], #1

CMP r0, #0

SWINE SWI\_write

BNE prntLoop

MOV r8, r3

nextletter LDRB r0, [r8], #1

CMP r0, #0

BNE Encrypt

B Increment

Encrypt MOV r9, r0

AND r9, r9, #0XFF000000 ;first byte masked--have to move to third byte------right shift 16

MOV r10, r0

AND r10, r10, #0X00FF0000 ;second byte masked-have to move to fourth byte-----right shift 16

MOV r11, r0

AND r11, r11, #0X0000FF00 ;third byte masked--have to move to first byte-------left shift 16

MOV r12, r0

AND r12, r12, #0X000000FF ;fourth byte masked-have to move to second byte------left shift 16

MOV r9, r9, LSR#16

MOV r10, r10, LSR#16

MOV r11, r11, LSL#16

MOV r12, r12, LSL#16

MOV r5, #0

ORR r5, r5, r9

ORR r5, r5, r10

ORR r5, r5, r11

ORR r5, r5, r12

EOR r5, r5, r4

;store r5 in array and duplicate array before printing

MOV r6, r5

Print MOV r10,#8 ;count of nibbles = 8

LOOP MOV r0,r6,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 r6,r6,LSL #4 ;shift left one nibble

SUBS r10,r10, #1 ;decrement nibble count

BNE LOOP ;if more nibbles,loop back

EOR r5, r5, r4

MOV r9, r5

AND r9, r9, #0XFF000000 ;first byte masked--have to move to third byte------right shift 16

MOV r10, r5

AND r10, r10, #0X00FF0000 ;second byte masked-have to move to fourth byte-----right shift 16

MOV r11, r5

AND r11, r11, #0X0000FF00 ;third byte masked--have to move to first byte-------left shift 16

MOV r12, r5

AND r12, r12, #0X000000FF ;fourth byte masked-have to move to second byte------left shift 16

MOV r9, r9, LSR#16

MOV r10, r10, LSR#16

MOV r11, r11, LSL#16

MOV r12, r12, LSL#16

MOV r5, #0

ORR r5, r5, r9

ORR r5, r5, r10

ORR r5, r5, r11

ORR r5, r5, r12

MOV r0, r5

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