

Task1

Task 3 code:

; Finite State Machine (Finite.asm)

; This program implements a finite state machine that

; accepts an integer with an optional leading sign.

INCLUDE Irvine32.inc

ENTER\_KEY = 13

.data

InvalidInputMsg BYTE "Invalid input",13,10,0

.code

main PROC

call Clrscr

StateA:

call Getnext ; read next char into AL

call IsDigit

jz StateC

call DisplayErrorMsg ; invalid input found

jmp Quit

StateB:

call Getnext ; read next char into AL

cmp al, ENTER\_KEY

je Quit

call DisplayErrorMsg ; invalid input found

jmp Quit

StateC:

call Getnext ; read next char into AL

call IsDigit ; ZF = 1 if AL contains a digit

; because what is efficency

jz StateC

cmp al,'a'

je StateC

cmp al,'A'

je StateC

cmp al,'b'

je StateC

cmp al,'B'

je StateC

cmp al,'c'

je StateC

cmp al,'C'

je StateC

cmp al,'d'

je StateC

cmp al,'D'

je StateC

cmp al,'e'

je StateC

cmp al,'E'

je StateC

cmp al,'f'

je StateC

cmp al,'F'

je StateC

cmp al,'h'

je StateB

call DisplayErrorMsg ; no: invalid input found

jmp Quit

Quit:

call Crlf

exit

main ENDP

;-----------------------------------------------

Getnext PROC

;

; Reads a character from standard input.

; Receives: nothing

; Returns: AL contains the character

;-----------------------------------------------

call ReadChar ; input from keyboard

call WriteChar ; echo on screen

ret

Getnext ENDP

;-----------------------------------------------

DisplayErrorMsg PROC

;

; Displays an error message indicating that

; the input stream contains illegal input.

; Receives: nothing.

; Returns: nothing

;-----------------------------------------------

push edx

mov edx,OFFSET InvalidInputMsg

call WriteString

pop edx

ret

DisplayErrorMsg ENDP

END main