**Lab 14 (Portal 14)**

**Aamir Sohail**

**18F-0215**

# **PROBLEM 1**

# PART 1

In real-address mode, an x86 processor can access 1,048,576 bytes of memory (1 MByte) using 20-bit addresses in the range 0 to FFFFF hexadecimal.

Protected mode is the more powerful “native” processor mode. When running in protected mode, a program’s linear address space is 4 GBytes, using addresses 0 to FFFFFFFF hexadecimal.

# PART 2

The code is correct final value of C is 1D.

# PART 3

mov eax,00401000

Jmp eax

# Part 4

mov al ,40

add al, 95

# PART 5

## Code:

INCLUDE irvine32.inc

.data

Array db 8 Dup(?)

.code

main proc

call Readdec

call writedec

mov ecx,8

mov edi,offset Array

mov dl,al

L1:

rol dl,1

jc \_1

\_0:

mov al,0

jmp goOn

\_1:

mov al,1

goOn:

stosb

Loop L1

mov esi,offset Array

mov ebx,type Array

mov ecx, sizeof Array

call dumpmem

exit

main endp

end main

## Screenshot:

# PART 6

The assembler will proceed at label l3. This is because the value is greater when we compare them and according to condition , l3 is to be run in case of greater value.

# Problem 02-A:

## Code:

INCLUDE irvine32.inc

.data

st1 db "Enter Character (Hex-input) : ",0

stA\_ db "State A :",0

stB\_ db "State B :",0

stC\_ db "State C :",0

.code

main proc

State\_A:

mov edx,offset stA\_

call writestring

call Crlf

call getnext

call IsDigit1

cmp ebx,0

jz State\_C

cmp ebx,1

jz State\_B

jmp State\_A

State\_B:

mov edx,offset stb\_

call writestring

call Crlf

call getnext

call IsDigit1

cmp ebx,0

jz State\_C

jmp State\_B

State\_C:

mov edx,offset stC\_

call writestring

call Crlf

call getnext

call IsDigit1

cmp ebx,0

jz State\_C

jmp State\_B

exit

main endp

;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;

Getnext proc

mov edx,offset st1

call writestring

call readchar

call Writechar

call crlf

ret

Getnext endp

;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;

IsDigit1 proc

cmp al,'0'

jb ERR

cmp al,'9'

ja hexcheck

mov ebx,0

jmp next

Hexcheck:

cmp al,'A'

jb ERR

cmp al,'F'

ja ERR

mov ebx,1

jmp next

ERR:

mov ebx,2

call DisplayErrorMsg

next:

ret

IsDigit1 endp

;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;

.Data

temp db "Input is not a digit nor integer !!! Input again";

.Code

DisplayErrorMsg proc

call crlf

mov edx,offset temp

call writestring

call crlf

ret

DisplayErrorMsg endp

end main

## Screenshot:

# Problem 02-B:

## Code:

INCLUDE irvine32.inc

.data

st1 db "Write Target String : ",0

st2 db 26 DUP(?),0

FrequencyTable db 255 DUP(0)

.code

main proc

call str\_in

call Get\_frequencies

call Print\_frequency

exit

main endp

;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;

str\_in proc

mov edx,offset st1

call writestring

mov edx,offset st2

mov ecx,sizeof st2

call readString

ret

str\_in endp

;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;

Get\_frequencies proc

mov ecx,26

mov esi,offset st2

mov edi,offset FrequencyTable

mov ebx,0

L1:

mov bl,[esi]

mov al,[edi+ebx]

inc al

mov [edi+ebx],al

inc esi

Loop l1

ret

Get\_frequencies endp

.Data

temp db "Index",0

temp2 db "Frequency",0

.Code

Print\_frequency proc

;;;;;;;;;;;;;;;;Display Count of Capital Alphabets;;;;;;;;;;;;

mov edx,offset temp

call writestring

mov eax," "

call Writechar

call Writechar

mov edx,offset temp2

call writestring

call crlf

mov esi,offset FrequencyTable

add esi,41h

mov ecx,26

Lp:

mov eax,5Bh

sub eax,ecx

call writeHex

mov eax," "

call Writechar

call Writechar

Lodsb

call writeDec

call crlf

Loop Lp

ret

Print\_frequency endp

end main

## Screenshot: