**Andrew Tran**

**Lab Experience Nine**

**Lab Objectives**

1. Utilizing the irvine32 library.

2. Understanding the Loop instruction in Assembler.

3. Utilizing the cmp and conditional jump instructions.

4. Implementing the logical operators and, or, not, and xor.

**Exercise 1**

Do problems 1, 4, 5,8,10 on pages 236-237 in your textbook. Place all answers in a word document.

**Exercise 2**

Write a complete program to prompt for and input a string and a single character.

The input string will be compressed by removing all occurrences of the character entered by the user.

As an example: Suppose the user entered the string “**The Life of Riley”** and the character **e**.

The new string will be “**Th Lif of Rily”** (The double quotes are not input by the user.).

After you have created the new compressed string (by altering the input string), display it to the monitor.

**NOTE: Creating a new string variable with the character removed is not an acceptable solution. You need to change the original string without using extra memory.**

You may assume the maximum length of the input string is less than or equal to 50 characters.

For full credit to be received, you should use a procedure to compress the string based on the character input. Your procedure should accept the address of the character and the address of the string.

Any reference to a data label defined in the data segment will result in a significant loss of possible points.

Capture the output window and paste the captured screen in the word document.

**What to hand in:**

1. Print outs of the assembler programs used to solve the lab exercises.

2. Compress all of the assembler files into a single file using your name and the lab number as the filename.

3. Place the compressed file into the D2L Dropbox folder titled Lab Experience Nine.

**Due Date: As indicated on the D2L Dropbox folder.**

Problems

1. And al, 00001111b
2. Cmp dx,cx

Jbe L1

1. Cmp ax,cx

Jg L2

8. ja L1

Cmp ebx,val1

Ja L1

Mov x,2

Jmp L2

L1:

Mov x,1

L2:

10. move ax,N

Beginwhile:

Cmp eax,0

Jng endwhile

Mov ebx,A

Mov ecx,b

Cmp n,3

Jne L1

Jmp L3

Jmp endwhile

L1:

Cmp N,ebx

Jl L2

Cmp N,ecx

Jg L2

Jmp L3

Jmp endwhile

L2:

Mov N,n-2

Jmp endwhile

L3:

Mov n,n-1

Jmp endwhile

Endwhile:

Mov N,eax

Exercise 2:

INCLUDE Irvine32.inc

.data

msg BYTE "String: ", 0

msg2 BYTE "Character: ", 0

string BYTE 51 DUP(0)

char BYTE ?

count DWORD ?

.code

main PROC

mov edx, OFFSET msg

call writeString

mov edx, OFFSET string

mov ecx, 50

call readString

mov count, eax

call crlf

mov edx, OFFSET msg2

call writeString

call readChar

mov char, al

call crlf

call crlf

mov ecx, count

mov edx, OFFSET string

call removal

mov edx, OFFSET string

call writeString

call crlf

main ENDP

removal PROC

mov eax, 0; keep a count of string w/o chosen char

L1:

movzx edx,string[ecx]

cmp dl, char; compare it to the letter

jz match; jump if match

push edx; otherwise push the non - chosen char onto stack

inc eax; inc the count w/o chosen char

match:

mov string[ecx], 0; replace with 0 if match

loop L1

mov ecx, eax; move count to ecx

mov eax, 0

L3:

pop edx; pop back all non - chosen char back

mov BYTE PTR string[eax], dl

inc eax

loop L3

ret

removal ENDP

END main

