Name: Umer Azeem

Project code:

.386

.model flat, stdcall

.stack 4096

INCLUDE irvine32.inc

ExitProcess PROTO, dwExitCode:DWORD

.data

NAMEMSG db "Enter your name: ", 0

passwordMSG db "Enter your password (at least 6 characters): ", 0

successfullylogin db "Welcome, you are logged in", 0

Failedlogin db "You entered an incorrect username or password.", 0

registrationMSG db "I am here to get you registered. Tell your new username and password.", 0

passwordShort db "Password is too short. It must be at least 6 characters.", 0

nameofuser db 32 dup(0)

password db 32 dup(0)

Userinput db 32 dup(0)

Passinput db 32 dup(0)

.code

main PROC

call Clrscr

mov edx, OFFSET registrationMSG

call WriteString

call Crlf

mov edx, OFFSET NAMEMSG

call WriteString

call Crlf

mov edx, OFFSET nameofuser

call ReadString

mov byte ptr [nameofuser + eax], 0

input\_password:

mov edx, OFFSET passwordMSG

call WriteString

call Crlf

mov edx, OFFSET password

call ReadString

mov byte ptr [password + eax], 0 ; Null-terminate the string

mov esi, OFFSET password

call StringLength

cmp eax, 6

jl password\_to\_short

mov edx, OFFSET NAMEMSG

call WriteString

call Crlf

mov edx, OFFSET Userinput

call ReadString

mov byte ptr [Userinput + eax], 0 ; Null-terminate the string

mov edx, OFFSET passwordMSG

call WriteString

call Crlf

mov edx, OFFSET Passinput

call ReadString

mov byte ptr [Passinput + eax], 0 ; Null-terminate the string

cmp byte ptr [Userinput], 0

je loginerror

cmp byte ptr [Passinput], 0

je loginerror

mov edx, OFFSET Userinput

mov ecx, OFFSET nameofuser

call StringCompare

jne loginerror

mov edx, OFFSET Passinput

mov ecx, OFFSET password

call StringCompare

jne loginerror

mov edx, OFFSET successfullylogin

call WriteString

jmp end\_P

loginerror:

mov edx, OFFSET Failedlogin

call WriteString

jmp end\_P

password\_to\_short:

mov edx, OFFSET passwordShort

call WriteString

call Crlf

jmp input\_password

end\_P:

mov edi, OFFSET nameofuser

mov ecx, sizeof nameofuser

call ZeroBuffer

mov edi, OFFSET password

mov ecx, sizeof password

call ZeroBuffer

; End program

INVOKE ExitProcess, 0

main ENDP

StringCompare PROC

push esi

push edi

mov esi, edx

mov edi, ecx

call StringLength

mov ebx, eax

mov esi, edx

call StringLength

cmp ebx, eax

jne stringnotequal

comploop:

mov al, [esi]

mov bl, [edi]

cmp al, bl

jne stringnotequal

test al, al

je stringsequal

inc esi

inc edi

jmp comploop

stringsequal:

xor eax, eax

jmp end\_compare

stringnotequal:

mov eax, 1

end\_compare:

pop edi

pop esi

ret

StringCompare ENDP

StringLength PROC

push esi

mov esi, edx

xor eax, eax

stringlength\_loop:

mov al, [esi]

test al, al

je stringlength\_done

inc esi

inc eax

jmp stringlength\_loop

stringlength\_done:

pop esi

ret

StringLength ENDP

ZeroBuffer PROC

xor eax, eax

rep stosb

ret

ZeroBuffer ENDP

END main