Assembly Language Programming Exercise  
Option 1

**Code:**

.586

.model flat, stdcall

option casemap :none

.stack 4096

extrn ExitProcess@4: proc

GetStdHandle proto :dword

ReadConsoleA proto :dword, :dword, :dword, :dword, :dword

WriteConsoleA proto :dword, :dword, :dword, :dword, :dword

MessageBoxA proto :dword, :dword, :dword, :dword

STD\_INPUT\_HANDLE equ -10

STD\_OUTPUT\_HANDLE equ -11

.data

intro\_string db "Enter temperature value:",0

convert\_string db "Change value to (C/F):",0

bufSize = 80

inputHandle DWORD ?

buffer db bufSize dup(?)

bytes\_read DWORD ?

sum\_string db "Temperature is:",0

outputHandle DWORD ?

bytes\_written dd ?

actualNumber dw 0

converter dw 0

asciiBuf db 4 dup (" ")

.code

main: ; main Procedure begins from here

invoke GetStdHandle, STD\_OUTPUT\_HANDLE ; Outputs the first line

mov outputHandle, eax

mov eax,LENGTHOF intro\_string ; length of intro\_string

invoke WriteConsoleA, outputHandle, addr intro\_string, eax, addr bytes\_written, 3

invoke GetStdHandle, STD\_INPUT\_HANDLE ; Reads user input

mov inputHandle, eax

invoke ReadConsoleA, inputHandle, addr buffer, bufSize, addr bytes\_read, 5

sub bytes\_read, 2 ; -2 to remove cr,lf

mov ebx,0

mov al, byte ptr buffer+[ebx]

sub al,30h

add [actualNumber],ax

getNext:

inc bx

cmp ebx,bytes\_read

jz cont

mov ax,10

mul [actualNumber]

mov actualNumber,ax

mov al, byte ptr buffer+[ebx]

sub al,30h

add actualNumber,ax

jmp getNext

; Converts user input to readable number

cont:

invoke GetStdHandle, STD\_OUTPUT\_HANDLE ; Prompts to the user to enter the conversion

mov outputHandle, eax

mov eax,LENGTHOF convert\_string ; length of convert\_string

invoke WriteConsoleA, outputHandle, addr convert\_string, eax, addr bytes\_written, 3

;------------------READING CONVERTER --------------------;

; Reads conversion character that the user entered

invoke GetStdHandle, STD\_INPUT\_HANDLE

mov inputHandle, eax

invoke ReadConsoleA, inputHandle, addr buffer, bufSize, addr bytes\_read, 5

sub bytes\_read, 2 ; -2 to remove cr,lf

mov ebx,0

mov al, byte ptr buffer+[ebx]

sub al,30h

add [converter],ax

CMP ax, 22 ; Checks for user input if it matches

jz centTOfarenFunct ; Then jumps to the certain procedure

CMP ax, 19

jz farenTOcentFunct

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

continue:

; Outputs to the console the line giving the temperature result

invoke GetStdHandle, STD\_OUTPUT\_HANDLE

mov outputHandle, eax

mov eax,LENGTHOF sum\_string ;length of sum\_string

invoke WriteConsoleA, outputHandle, addr sum\_string, eax, addr bytes\_written, 3

mov ax,[actualNumber]

mov cl,10

mov ebx,3

nextNum:

div cl

add ah,30h

mov byte ptr asciiBuf+[ebx],ah

dec ebx

mov ah,0

cmp al,0

ja nextNum

mov eax,4

; Outputs to the console the converted temperature

invoke WriteConsoleA, outputHandle, addr asciiBuf, eax, addr bytes\_written, 0

; Outputs a message box with the converted temperature

invoke MessageBoxA, 0, addr asciiBuf, addr sum\_string,16

mov eax,0

mov eax,bytes\_written

push 0

call ExitProcess@4 ; Calls the exit process and ends the program

centTOfarenFunct:

call centTOfaren

jmp continue

centTOfaren PROC

mov ax,9

mov cx, [actualNumber] ; cx = [actualNumber]

mul cx ; ax = ax \* cx

mov [actualNumber],ax

mov ax, [actualNumber] ; ax = [actualNumber]

mov cx, 5 ; cx = 5

div cx ; ax = ax / cx

add ax,32

mov [actualNumber],ax

ret

centTOfaren ENDP

farenTOcentFunct:

call farenTOcent

jmp continue

farenTOcent PROC

mov ax,[actualNumber]

sub ax,32

mov cx, 5 ; cx = 5

mul cx ; ax = ax \* cx

mov cx, 9 ; cx = 9

div cx ; ax = ax / cx

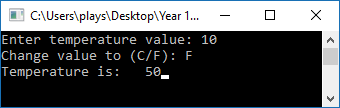
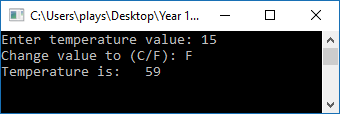
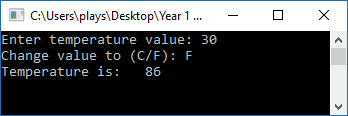
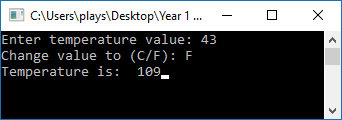
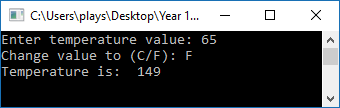
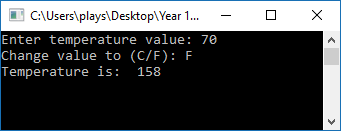
mov [actualNumber],ax

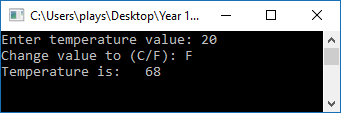
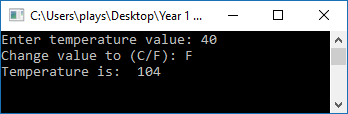
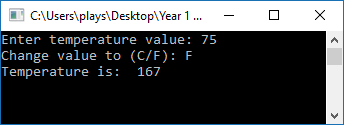
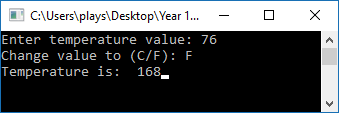
ret

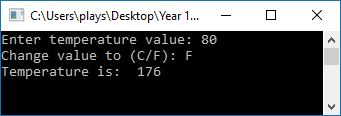
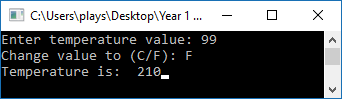
farenTOcent ENDP

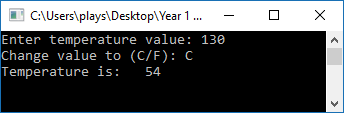
end main

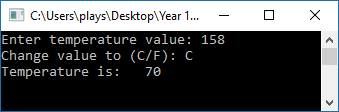
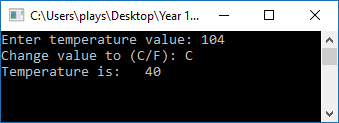
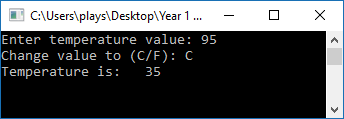
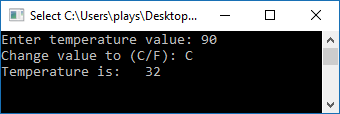
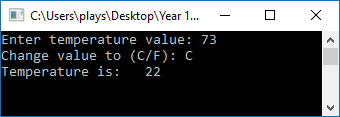
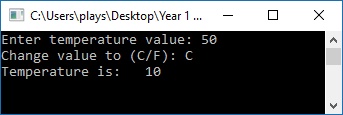
**Screenshots:**

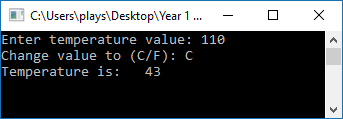
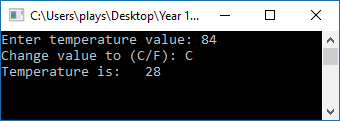
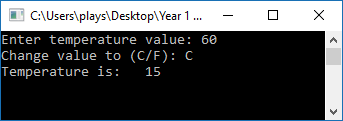
**Celsius to Fahrenheit**

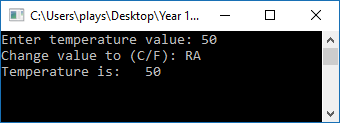




**Fahrenheit to Celsius**





**Invalid Entries**

