National University of Computer and Emerging Sciences



Lab 05

COAL

|  |  |
| --- | --- |
| Name | Muhammad Zain |
| Roll No. | 19F-0228 |
| LAB INSTRUCTOR | Mr Abuzar Ghafari |
| Semester | Fall 2020 |

Task 1

# **Source Code**

*; Author: Muhammad Zain*

; Program Name: Task 1 letter grading

; Program Description: .IF directive example

; Date: 04/11/2000

INCLUDE Irvine32.inc

.data

;==============================================================

;=====Grading Policy===========================================

Letter\_grade1 byte " Grade A",0 ;90 to 100 A

Letter\_grade2 byte " Grade B",0 ;80 to 89 B

Letter\_grade3 byte " Grade C",0 ;70 to 79 C

Letter\_grade4 byte " Grade D",0 ;60 to 69 D

Letter\_grade5 byte " Grade F",0 ;0 to 59 F

;=============================================================

;=======Validation Messages===================================

Input\_Msg byte " Input your Number between 0 to 100 to get the Grade",0

Invalid\_Msg byte " Invalid Input ,Please input again ",0

;==============================================================

.code

main PROC

mov edx,offset Input\_Msg

call writeString

call crlf

call readint

;=============Validation loop==================================

.while (eax < 0 || eax >100)

mov edx,offset Invalid\_Msg

call writeString

call crlf

call readint

.endw

;===============================================================

.if (eax >89);================== A grade

mov edx,offset letter\_grade1

call writeString

.elseif (eax >79);============= B grade

mov edx,offset letter\_grade2

call writeString

.elseif (eax >69);============= C grade

mov edx,offset letter\_grade3

call writeString

.elseif (eax >59);============= D grade

mov edx,offset letter\_grade4

call writeString

.else ;=====================F grade

mov edx,offset letter\_grade5

call writeString

.endif

call crlf

call WaitMsg

exit

main ENDP

END main

# **Snip**

A screen shot of a computer

Description automatically generatedText

Description automatically generatedText

Description automatically generatedText

Description automatically generatedText

Description automatically generated

Task 2

# **Source Code**

*; Author: Muhammad Zain*

; Program Name: Task 2 gates

; Program Description: gates

; Date: 04/11/2000

INCLUDE Irvine32.inc

.data

;=============== menu ===============================

menu\_1 byte "Enter 1 for AND Operation ", 0

menu\_2 byte "Enter 2 for OR Operation ", 0

menu\_3 byte "Enter 3 for NOT Operation ", 0

menu\_4 byte "Enter 4 for XOR Operation ", 0

menu\_5 byte "Enter 5 to EXIT ", 0

;========================================================

;=========== Input Values================================

value1 db "Enter Value\_1 = ",0

value2 db "Enter Value\_2 = ",0

Answer dword ?

;========= validation ====================================

Validation\_msg byte "Invalid Input enter again ",0

;=========================================================

enter\_again byte " Input again from the above menu ", 0

;========================================================

.code

main PROC

;.REPEAT

mov edx,offset menu\_1 ;======= And operation message

call writestring

call crlf

mov edx,offset menu\_2 ;======= or operation message

call writestring

call crlf

mov edx,offset menu\_3 ;======= xor operation message

call writestring

call crlf

mov edx,offset menu\_4 ;======= not operation message

call writestring

call crlf

mov edx,offset menu\_5 ;======= exit operation message

call writestring

call crlf

;=======Taking Input and validation ========================================

call readint

.while(eax > 5 || eax <1)

mov edx,offset Validation\_msg

call writestring

call readint

.endw

;============================================================================

;============================================================================

.while (eax !=5)

.if (eax ==1)

call AND\_OP

mov Answer,eax

call writehex

.elseif (eax ==2)

call OR\_OP

mov Answer,eax

call writehex

.elseif(eax ==3)

call NOT\_OP

mov Answer,eax

call writehex

.elseif (eax ==4)

call XOR\_OP

mov Answer,eax

call writehex

.endif

call crlf

mov edx,offset enter\_again

call writestring

call crlf

call readint

.endw

exit

main ENDP

;================================= And Function =================================

AND\_OP PROC

call crlf

mov edx,offset value1

call writestring

call readhex

mov ebx,eax

mov edx,offset value2

call writestring

call readhex

AND eax,ebx

ret

AND\_OP ENDP

;================================= OR Function =================================

OR\_OP PROc

call crlf

mov edx,offset value1

call writestring

call readhex

mov ebx,eax

mov edx,offset value2

call writestring

call readhex

OR eax,ebx

ret

call crlf

OR\_OP ENDP

;================================= NOt Function =================================

NOT\_OP PROC

call crlf

mov edx,offset value1

call writestring

call readhex

NOT eax

ret

call crlf

NOT\_OP ENDP

;================================= xor Function =================================

XOR\_OP PROC

call crlf

mov edx,offset value1

call writestring

call readhex

mov ebx,eax

mov edx,offset value2

call writestring

call readhex

XOR eax,ebx

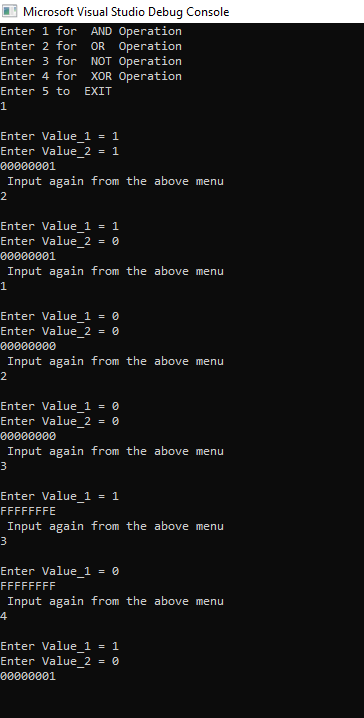
ret

call crlf

XOR\_OP ENDP

END main

# **Snip:**



Task 3

# **Source Code**

*; Author: Muhammad Zain*

; Program Name: Task 3

; Program Description: palindrome

; Date: 04/11/2000

INCLUDE Irvine32.inc

.data

prompt1 byte "Input the string to check wheter it is Palindrome or not ",0

prompt2 byte "yes, it is Palindrome",0

prompt3 byte "No, it is not Palindrome",0

;=============================================================

;=============================================================

;Initializing Array with 0 and reverse array to match

getString byte 50 dup (0)

reverseString byte 50 dup(? )

stringCounter dword ?

;bool yes or no

is\_Palindrome byte ?

TRUE = 1

FALSE = 0

.code

main PROC

;print main message

mov edx,offset prompt1

call writestring

call crlf

;=============================================================

;=============================================================

;taking input of string

mov edx,offset getString

mov ecx,sizeof getString

call readstring

mov stringCounter ,eax

mov ecx,stringCounter

;=============================================================

;=============================================================

;pushing the main string in stack

mov esi ,0

L1:

movzx eax,getString[esi]

push eax

inc esi

loop L1

;=============================================================

;=============================================================

;Pushing reverse string

mov esi ,0

mov ecx,stringCounter

l2:

pop eax

mov reverseString[esi], al

inc esi

loop l2

;calling function

call check\_Palindrome

;checking values

.if( is\_Palindrome== FALSE)

mov edx,offset prompt3

call writestring

call crlf

.else

mov edx,offset prompt2

call writestring

call crlf

.endif

exit

main ENDP

;=============================================================

;=============================================================

;first we take the user string

;then we push in stack for the sake of reversing

;then we reverse it

;save it another string

;compare both string

check\_Palindrome PROC

mov esi, 0

mov ecx, stringCounter

mov is\_Palindrome, TRUE

checkLoop:

mov al,reverseString[esi]

.if(getString[esi]==al)

mov is\_Palindrome,TRUE

.else

mov is\_Palindrome,FALSE

inc esi

.endif

ret

check\_Palindrome ENDP

END mainENDP

END main

# **Snip**

Text

Description automatically generatedText

Description automatically generated