Muhammad iftikhar

Section A

Roll Number [p180054@nu.edu.pk](mailto:p180054@nu.edu.pk)

include irvine32.inc

.data

val byte 65

msg byte "A ",0

msg1 byte "B ",0

msg2 byte "C",0

msg3 byte "D",0

msg4 byte "F",0

mss byte "invalid",0

.code

main proc

mov edx,offset msg

xor eax,eax

mov al,val

cmp al,100

JB L1

;jum below if al < 100

L1:

cmp al,90

JA L2

cmp al,80

JA L3

cmp al,70

JA L5

cmp al,60

JA L7

cmp al,0

JA L9

L2: ;jump above is al>90

mov edx,offset msg

call writestring

call crlf

L3:

cmp al,90

JB L4

L4:

mov edx,offset msg1

call writestring

call crlf

L5 :

cmp al,80

JB L6

L6:

mov edx,offset msg2

call writestring

call crlf

L7:

mov edx,offset msg3

call writestring

call crlf

L9:

cmp al ,60

JB L10

L10:

mov edx,offset msg4

call writestring

call crlf

main endp

end main

Question Number 2

include irvine32.inc

.data

array byte 10 dup(0)

count Dword ?

val dword ?

.code

main proc

;clearing

xor ecx,ecx

xor eax,eax

xor esi,esi

mov ebx,lengthof array

mov val,ebx

mov ecx,ebx ;lenght of the list

L1:

mov count ,ecx

mov ecx,al-1

L2:

mov bl,array[esi]

cmp array[esi] ,array[esi+1]

JA L3

xchg array[esi],array[esi+1]

xchg array[esi+1],array[esi]

loop L2

mov ecx,count

loop L1

exit

main endp

end main

Include irvine32.inc

.data

a byte 16

b byte 32

com byte 64

d byte 64

equation byte ?

ls byte ?

gr byte ?

e byte ?

count byte 0

h1 byte 0

h2 byte 0

h3 byte 0

.code

main PROC

;clearing

xor eax,eax

xor ebx,ebx

xor ecx,ecx

xor edx,edx

mov al,a

add al,b

mov e,al ; e= a+b

mov al,e

cmp al,d ;if e==d

jz next ;if e==d then zero flag is set the goto the next

mov equation ,1

next:

jNz l1 ;jump is zero falf is not zero

mov equation ,0

l1:

cmp al,d ;if e>d then zero flag and carry falg is zero

Ja l9 ;JA check both CF,Zf if both zero then return True

mov gr,1

l9:

JNBE l3 ;if ZF and Cf is not Equal

mov gr,0 ; gr=0

l3:

cmp al,d ;if (al<5) then carry falg is set

JC l4 ;if carry flag is set then do l4

mov ls,1

l4:

;else

JNC l5 ;if not less then

mov ls,0

beginwhile:

cmp al,d

JNE endwhile ;if not equal

inc eax

inc count

call writedec

call crlf

jmp beginwhile

endwhile:

mov d,al

l5:

;Quesiton 2 part !!

exit

exit

main ENDP

END main