**ID: 18K-0166**

**Name: Zaeem Ahmed**

**Section: A**

**Lab Tasks: 2 to 6**

**Lab 2**

**CODE**

TITLE My First Program (Test.asm)

INCLUDE Irvine32.inc

.data

;task1

val1 SBYTE ?

val2 SBYTE -10h

;task2

idata WORD ?

;task3

val3 SDWORD 12,147,483,648h

;task4

wArray BYTE 1h,2h,3h

;task5

myCol BYTE "Black",0

;task6

A DWORD 12

B DWORD 2

val DWORD 1

D DWORD 8

E DWORD 14

value DWORD ?

.code

main PROC

;task1

mov al,val2

mov val1,al

mov ah,val1

;task2

mov idata,bx

mov cx,idata

;task3

mov edx,val3

;task 4

mov eax,ASS

mul B

mov ecx,val

add eax,ecx

mul D

sub eax,E

mov value,eax

mov ebx,value

call DumpRegs

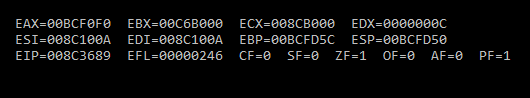
exit

main ENDP

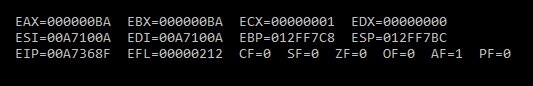
END main

**SNIPPETS**

**Q1-Q5**



**Q6**



Lab 3

Q1

.data

x byte 10h

y byte 20h

.code

MOV al,x

inc al

mov ah,y

dec ah

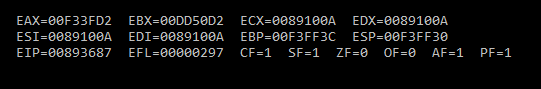
add ah,y

sub al,ah

mov x,al

mov bl,x

call DumpRegs



Q2

.data

val1 word 8h

val2 word 15h

val3 word 20h

.code

neg val2

mov ax,val2

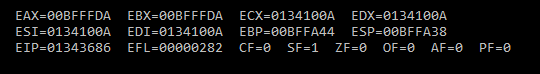
add ax,7h

sub ax,val3

add ax,val1

mov ebx,eax

call DumpRegs



Q3

.data

len word 8h

area word ?

.code

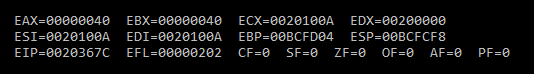
movzx eax,len

mul ax

mov area,ax

movzx ebx,area

call DumpRegs



Q4

.data

len word 8h

wid word 9h

area word ?

.code

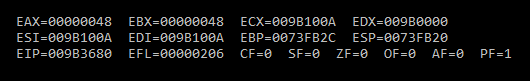
movzx eax,len

mul wid

mov area,ax

movzx ebx,area

call DumpRegs



Q5

.data

h word 8h

alt word 15h

area word ?

a byte 2h

.code

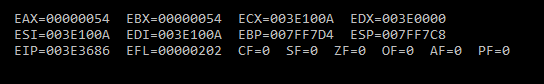
movzx eax,h

mul alt

div a

mov area,ax

movzx ebx,area



Q6

.data

val1 BYTE 10h

val2 WORD 8000h

val3 DWORD 0FFFFh

val4 WORD 7FFFh

.code

main PROC

;i

inc val2

movzx ebx,val2

;ii

sub eax,val3

;iii

movzx ecx,bx

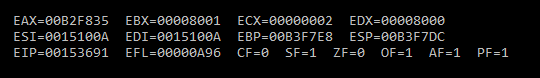
sub cx,val4

;v

inc val4

movzx edx,val4

call DumpRegs



Lab 4

Q1

.data

arrayB BYTE 5h, 6h, 2h

arrayW WORD 15h, 5h, 10h

arrayD DWORD 60h, 12h, 18h

prod1 DWORD ?

prod2 DWORD ?

prod3 DWORD ?

.code

;PROD1 = arrayB[0] \* arrayW[0] \* arrayD[0]

movzx eax,arrayB[0]

mul arrayW[0]

mul arrayD[0]

mov ebx,eax

;PROD2 = arrayB[1] \* arrayW[1] \* arrayD[1]

movzx eax,arrayB[1]

mul arrayW[2]

mul arrayD[4]

mov ecx,eax

;PROD3 = arrayB[2] \* arrayW[2] \* arrayD[2]

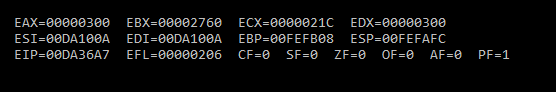
movzx eax,arrayB[2]

mul arrayW[4]

mul arrayD[8]

mov edx,eax

call DumpRegs



Q2

.data

u sbyte 5h

v sbyte 4h

w sbyte 8h

x sbyte 9h

y sbyte 6h

z sbyte ?

.code

;z = x + y + w – v +u

mov al,x

add al,y

add al,w

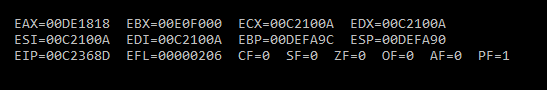
sub al,v

add al,u

mov z,al

mov ah,z

call DumpRegs



Q3

.data

array1 BYTE 10, 20, 30, 40

array2 WORD 4 DUP(?)

.code

mov esi,3

mov ecx,4

mov edi,0

L1:

movzx ax,array1[esi]

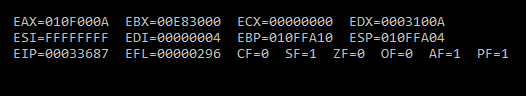
mov array2[edi],ax

inc edi

dec esi

loop L1

call DumpRegs



Lab 5

Q1

.data

array1 BYTE 10, 20, 30, 40, 50

.code

mov esi,0

mov ecx,lengthof array1

mov edi,sizeof array1

L1:

movzx eax,array1[esi]

push eax

add esi,type array1

loop L1

mov esi,0

mov ecx,lengthof array1

L2:

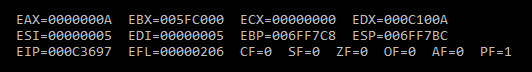
pop eax

mov array1[esi],al

add esi,type array1

loop L2

call DumpRegs



Q2

.code

main PROC

mov eax,1

call DumpRegs

mov ebx,0 ; initial setup

mov edx,1

mov ecx,9 ; count

L1:

mov eax,ebx ; eax = ebx + edx

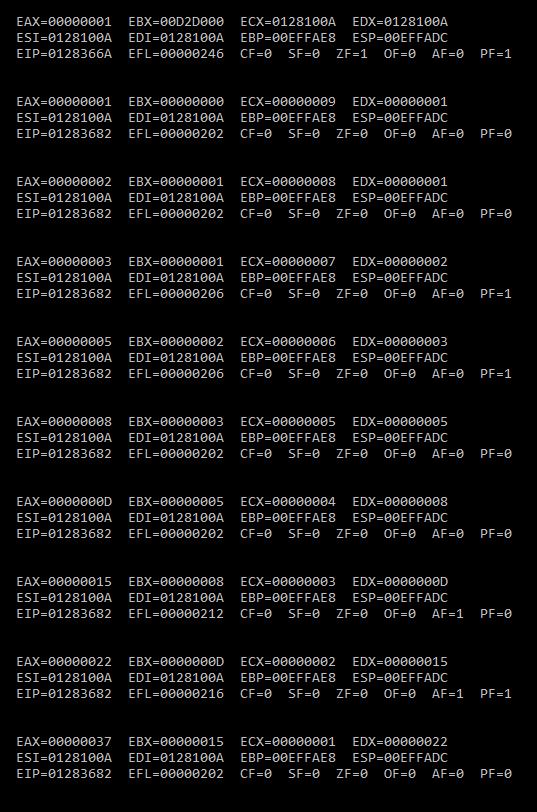
add eax,edx

call DumpRegs ; display eax

mov ebx,edx

mov edx,eax

Loop L1



Q3

.data

myArray BYTE 15, 10, 25, 20, 30

var byte ? ;used to swap

var1 dword 0 ; flag variable (to avoid error)

var2 sdword 1 ; flag variable (to avoid error)

.code

main PROC

; initialing registers

mov ebx,0

mov eax,0

mov esi,0

mov ecx,LENGTHOF myarray

l:

mov eax,ecx ;j

mov ebx,ecx ;i

dec eax ; j = i - 1

mov ecx , eax

;dec eax

cmp var2 ,ecx ; to aoid infinte loop when ecx = 0

jg m

l2:

mov dl , myarray[ebx\* TYPE myarray - 1] ; value at [i]

mov var,dl

mov dl,myarray[eax\* TYPE myarray - 1 ] ; value at [i-1]

cmp dl,var

jg l1 ;jump to swap

jmp x ; to avoid using swap algo

; SWAP

l1:

;dec ecx

mov myarray[ebx\* TYPE myarray -1],dl

mov dl,var

mov myarray[eax\* TYPE myarray - 1],dl

jmp x ; useless

x:

dec eax ; j = j - 1

loop l2

;cmp var1,ebx TO avoid error but seem useless

;jg m

mov ecx,ebx ; retrive i

loop l

m:

mov esi,0

mov ecx,5

mov dh,1

mov dl,0

print:

mov al,myarray[esi]

call writeint

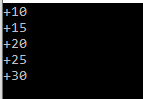
call gotoxy

inc esi

inc dh

mov dl,0

LOOP print



Lab 6

Q1

CR=0Dh

LF= 0Ah

.data

rand1 BYTE "Generating 20 random integers between "

BYTE "0 and 990:",CR,LF,0

.code

main PROC

;set text color to black text on green background:

mov eax,black + (green\*16)

call SetTextColor

call Clrscr

call Randomize

;"Generating 20 random integers between 0 and 999"

;put a delay between each

mov edx,OFFSET rand1

call WriteString

mov ecx,20

mov dh,2

mov dl,0

L1: call Gotoxy

mov eax,1000

call RandomRange

call WriteDec

mov eax,5

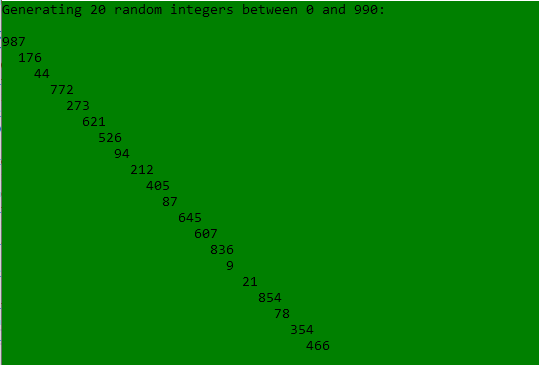
call Delay

inc dh

add dl,2

loop L1

call Crlf



Q2

.data

eId BYTE "Enter Your ID : ",0

eName BYTE "Enter Your Name : ",0

eYOB BYTE "Enter you year of birth : ",0

eSal BYTE "Enter Your Annual salary : ",0

id BYTE 20 DUP(?)

name1 BYTE 20 DUP(?)

yob word ?

salary dword ?

tr BYTE "Tax : ",0

fl BYTE "No Tax applicable.",0

.code

main PROC

mov edx, offset eId

call writeString

mov edx, offset id

mov ecx, 20

call readstring

mov edx, offset eName

call writeString

mov edx, offset name1

mov ecx, 20

call readstring

mov edx, offset eYOB

call writeString

mov edx, offset yob

call readDec

mov edx, offset eSal

call writeString

mov edx, offset salary

call readDec

mov ebx,50000d

cmp eax,ebx

JAE Greater

JMP notGreater

Greater:

mov esi, 2

mov edx, 0

div esi

mov edx, offset tr

call writeString

call writeDec

call crlf

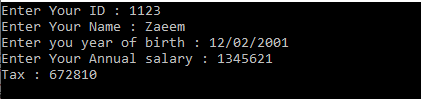
JMP exit\_pro

notGreater:

mov edx,offset fl

call writestring

exit\_pro:



Q3

.code

main PROC

mov eax,1

call DumpRegs

mov ebx,0 ; initial setup

mov edx,1

mov ecx,7 ; count

L1:

mov eax,ebx ; eax = ebx + edx

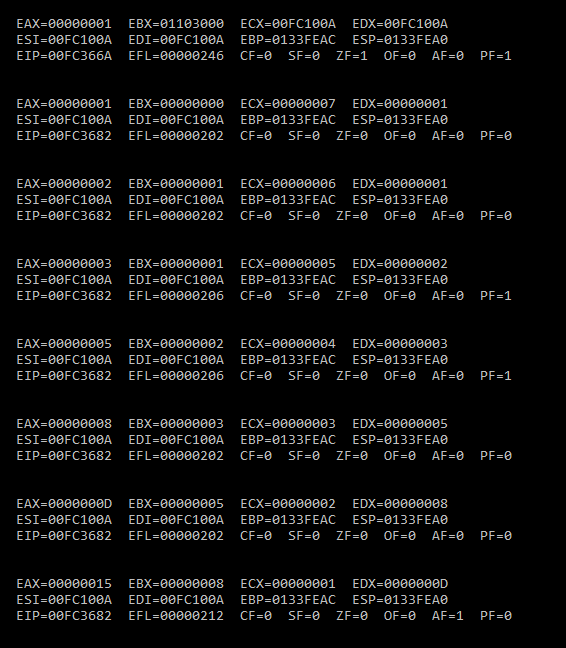
add eax,edx

call DumpRegs ; display eax

mov ebx,edx

mov edx,eax

Loop L1



Q4

.data

x dword '\*'

a BYTE 1

b BYTE 5

c1 BYTE 1

.code

main PROC

mov ecx,5

L1:

mov dl,b

mov dh,a

call Gotoxy

mov ebx,ecx

movzx ecx,c1

L2:

mov edx,OFFSET x

call WriteString

LOOP L2

mov ecx,ebx

inc c1

inc a

dec b

call crlf

Loop L1

