**COAL ASSIGNMENT 2**

**NAME : SUBHAN**

**NU-ID: 22K-4316**

**SECTION : 3F-BCS**

**Question 1:**

2. ZF = 0

4. ZF = 0

**Question 2:**

TITLE My First Program (Test.asm)

INCLUDE Irvine32.inc

.data

array1 sdword 40 , -90, -67 , 98 , 78 , -45 , 0 , 32

array2 sdword lengthof array1 DUP(?)

.code

storePosValues PROC, siz: DWORD, arr1: PTR SDWORD, arr2: PTR SDWORD

LOCAL index: DWORD

LOCAL index1: DWORD

push edi

push esi

mov esi , arr1

mov edi, arr2

push ecx

mov ecx, siz

push eax

mov index, 0

mov index1 , 0

l1:

mov edx, index

mov eax, [esi + edx]

cmp eax, 0

JLE \_else

\_if:

mov edx, index1

mov [edi + edx] , eax

add index1, 4

call writeInt

\_else:

add index, 4

sub ecx , 3

loop l1

pop eax

pop ecx

pop esi

pop edi

ret

storePosValues ENDP

main PROC

push eax

mov eax, sizeof array1

invoke storePosValues, eax, ADDR array1, ADDR array2

call DumpRegs

exit

main ENDP

END main

**Question 3:**

TITLE My First Program (Test.asm)

INCLUDE Irvine32.inc

.data

N DWORD 12

A DWORD 5

B DWORD 7

.code

main PROC

\_while:

mov eax, N

cmp eax, 0

JL \_end

\_if:

cmp eax , 3

JNE \_else

cmp eax, A

JL \_do

cmp eax, B

JLE \_else

\_do:

sub eax, 2

mov N, eax

JMP \_while

\_else:

sub eax, 1

mov N, eax

JMP \_while

\_end:

call DumpRegs

exit

main ENDP

END main

**Question 4:**

TITLE My First Program (Test.asm)

INCLUDE Irvine32.inc

.data

showO BYTE "o" , 0

showE BYTE "e" , 0

.code

main PROC

call ReadDec

cmp al, 1

JE \_do

cmp al, 3

JE \_do

cmp al, 2

JE \_other

cmp al, 4

JE \_other

JMP \_end

\_do:

mov edx, offset showO

call writeString

JMP \_end

\_other:

mov edx, offset showE

call writeString

\_end:

call DumpRegs

exit

main ENDP

END main

**Question 5:**

TITLE My First Program (Test.asm)

INCLUDE Irvine32.inc

.data

a DWORD 100

b DWORD 200

cc DWORD ? ; using cc instead of c because c is a reserved keyword in assembly

i DWORD ?

j DWORD ?

.code

main PROC

mov i, 5

\_for:

cmp i,0

JBE \_end

mov eax, a

add eax, b

mov b, eax

call writeDec

call crlf

mov j, 5

\_innerfor:

cmp j,0

JBE \_innerend

mov eax, a

sub eax, 1

mov a, eax

mov eax, cc

add eax, 10

mov cc, eax

sub j, 1

JMP \_innerfor

\_innerend:

mov eax, a

call writeDec

call crlf

mov eax, cc

call writeDec

call crlf

sub i, 1

JMP \_for

\_end:

call DumpRegs

exit

main ENDP

END main

**Question 6:**

TITLE My First Program (Test.asm)

INCLUDE Irvine32.inc

.data

alphabets BYTE "ABCDEFGHIJKLMNOPQRSTUVWXYZ" , 0

star BYTE "\*" , 0

space BYTE " " , 0

n DWORD ?

.code

main PROC

mov esi , 0

call readDec

mov ecx, eax

mov n, eax

mov eax, 1

l1:

push ecx

mov ebx, ecx

mov ecx, n

mov edx, offset star

l2:

call writeString

loop l2

mov ecx, ebx

sub ecx, 1

mov edx, offset space

l7:

cmp ecx, 0

JE \_l7end

call writeString

dec ecx

JMP l7

\_l7end:

mov ecx, eax

push eax

l3:

mov eax, 0

mov al, alphabets[esi]

call writeChar

add esi, 1

mov eax, lengthOf alphabets

sub eax, 1

cmp esi, eax

JNE \_else

\_if:

mov esi, 0

\_else:

loop l3

mov ecx, ebx

sub ecx, 1

mov edx, offset space

l9:

cmp ecx, 0

JE \_l9End

call writeString

dec ecx

JMP l9

\_l9END:

pop eax

add eax, 2

mov ecx, n

mov edx, offset star

l4:

call writeString

loop l4

pop ecx

call crlf

dec ecx

cmp ecx, 0

JNE l1

call DumpRegs

exit

main ENDP

END main

**Question 7:**

TITLE My First Program (Test.asm)

INCLUDE Irvine32.inc

.data

type1 BYTE "Please type 1 for first\_class" , 0

type2 BYTE "Please type 2 for economy\_class" , 0

seats BYTE 10 DUP(?)

fc BYTE 0

ec BYTE 5

ask1 BYTE "If its acceptable to be placed in the economy section: " , 0

ask2 BYTE "if its acceptable to be placed in first class section" , 0

ask3 BYTE "Next flight leaves in 3 hours" , 0

textE BYTE "Economy Class" , 0

textF BYTE "First class " , 0

yes BYTE "Enter 1 for yes" , 0

.code

main PROC

mov ecx, lengthOf seats

l1:

mov edx, offset type1

call writeString

call crlf

mov edx, offset type2

call writeString

call crlf

call readDec

call crlf

cmp eax , 1

JE \_if

cmp eax, 2

JE \_else

JMP \_end

\_if:

movzx ebx, fC

cmp ebx , 5

JGE \_innerif

\_do:

mov esi , ebx

mov seats[esi] , 1

mov eax, esi

add eax, 1

call writeDec

call crlf

mov edx, offset textF

call writeString

call crlf

add fC, 1

JMP \_end

\_innerif:

mov edx, offset ask1

call writeString

call crlf

mov edx, offset yes

call writeString

call crlf

call readDec

call crlf

cmp eax, 1

JE \_doElse

mov edx, offset ask3

call writeString

call crlf

JMP \_end

\_else:

movzx ebx , eC

mov edx, lengthOf seats

cmp ebx, edx

JGE \_elseinnerif

\_doElse:

mov esi , ebx

mov seats[esi] , 1

mov eax, esi

add eax, 1

call writeDec

call crlf

mov edx, offset textE

call writeString

call crlf

add eC, 1

JMP \_end

\_elseinnerif:

mov edx, offset ask2

call writeString

call crlf

mov edx, offset yes

call writeString

call crlf

call readDec

call crlf

cmp eax, 1

JE \_do

mov edx, offset ask3

call writeString

call crlf

JMP \_end

\_end:

dec ecx

cmp ecx, 0

JNE l1

call DumpRegs

exit

main ENDP

END main

**Question 8:**

TITLE My First Program (Test.asm)

INCLUDE Irvine32.inc

.data

list DWORD 5, 23, 45 , 76 , 78 , 2 , 9 , 12 , 97 , 3

msgFound byte "Number found in the list or array",0

msgNFound byte "Sorry, Number not found in the list or array",0

.code

main PROC

call readDec

mov ecx, lengthOf list

mov esi , 0

l1:

mov ebx , list[esi]

cmp ebx, eax

JNE \_else

\_if:

mov edx, offset msgFound

call writeString

call crlf

call writeDec

call crlf

mov eax, esi

call writeDec

call crlf

JMP \_end

\_else :

add esi , 4

loop l1

mov edx, offset msgNFound

call writeString

\_end:

call DumpRegs

exit

main ENDP

END main