QUESTION NO: 4 (9)

include Irvine32.inc

.data

msg BYTE "FAST NATIONAL UNIVERSITY",0

assci BYTE 20 DUP(?)

.code

main PROC

mov esi , OFFSET msg

mov ecx, lengthof msg

mov edi, offset assci

L1:

LODSB

add al,32

stosb

loop L1

mov ecx , lengthof assci

mov esi , offset assci

L2:

mov eax , 0

lodsb

call writehex

call crlf

loop L2

ext:

call dumpregs

exit

main ENDP

END main



QUESTION NO: 4(8)

include Irvine32.inc

.data

unSortedArray DWORD 100 dup(5)

searchElemnts DWORD 20 dup(2)

msg BYTE "not found",0

msg1 BYTE "found",0

count DWORD ?

.code

main PROC

mov edi , offset unSortedArray

mov esi , offset searchElemnts

mov ecx, 20

cld

L1:

mov edi , offset unSortedArray

mov eax , [esi]

repne scasd

jnz L3

mov edx , offset msg1

call writestring

L4: add esi, 4

loop L1

jmp ext

L3:

mov edx , offset msg

call writestring

mov eax , [esi]

call writedec

call crlf

jmp L4

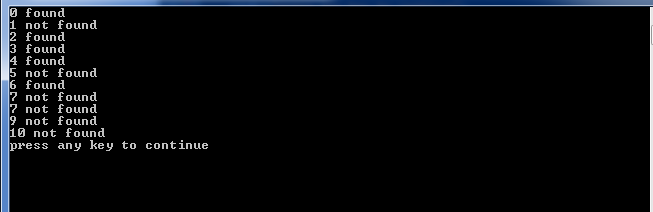
ext:

call dumpregs

exit

main ENDP

END main



QUESTION NO: 4(6)

include Irvine32.inc

.data

moon BYTE 20 DUP(?)

.code

main PROC

push offset moon

call star\_array

ext:

call dumpregs

exit

main ENDP

star\_array PROC

push ebp

mov ebp,esp

sub esp ,20

lea esi , [ebp-20]

mov edi , [ebp + 8]

mov ecx , 20

L1:

mov BYTE PTR [esi], 'x'

mov BYTE PTR [edi], '\*'

inc esi

inc edi

loop L1

mov ecx , 20

lea esi , [ebp-20]

L2:

mov eax , 0

lodsb

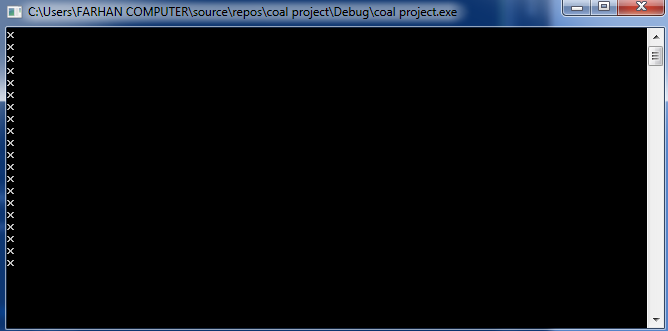
call writechar

call crlf

loop L2

star\_array endp

END main



Question no: 4(5)

include Irvine32.inc

.data

array DWORD 1,2,3,4,5,6,7,8,9,10

msg BYTE "not found",0

msg1 BYTE "found",0

.code

recursive proto, parray:PTR DWORD,val: DWORD

main PROC

INVOKE recursive , addr array , 2

ext:

call dumpregs

exit

main ENDP

recursive PROC,parray:PTR DWORD,val: DWORD

push ebp

mov ebp,esp

mov esi , parray

mov eax , val

L1:

cmp eax,[esi]

je L2

add esi,4

loop L1

mov edx , offset msg1

call writestring

jmp ext

L2:

mov edx , offset msg

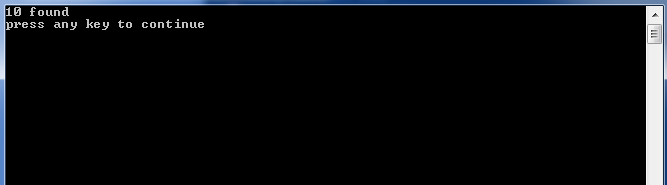
call writestring

ext:

ret

recursive endp

END main



QUESTION NO 4(7)

include Irvine32.inc

.data

.code

main PROC

push D4A4h

push 0Ah

call divider

ext:

call dumpregs

exit

main ENDP

divider proc

push ebp

mov ebp,esp

mov eax,[ebp + 12]

mov ebx,[ebp + 8]

div ebx

cmp eax,5h

jle hy

push eax

push ebx

call divider

hy:

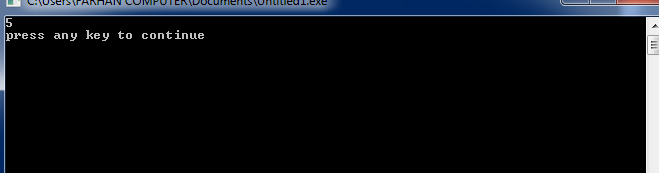
mov esp,ebp

pop ebp

ret 8

divider endp

END main



QUESTION NO: 4(1)

include Irvine32.inc

.data

sensor\_data dword ?

status DWORD ?

rev\_cnt dword ?

seq\_no dword ?

.code

main PROC

mov eax , 1111111111111111111111111111b

shld sensor\_data,eax,16

shl eax,16

shld status,eax,1

shld rev\_cnt , eax,3

shl eax,3

shld seq\_no,eax,12

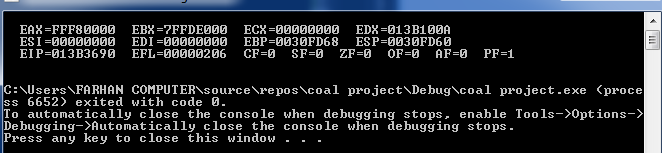
ext:

call dumpregs

exit

main ENDP

END main



QUESTION NO: 4(2)

INCLUDE IRVINE32.inc

.data

a dword ?

.code

main PROC

mov eax,0

call readdec

mov a,eax

mov ebx,eax

shl eax,4

add ebx,eax

mov eax,a

shl eax,2

add ebx,eax

mov eax,a

shl eax,1

add ebx,eax

mov eax,ebx

call writedec

exit

main ENDP

END main

