23k-2001
BCS-3J
Date ____ 20__

COAL : Assignment #02

Q1:

a.

EAX = 0000 000Dh , ECX = 0000 00DDh , EDX = 0000 ABCDh, ESP = 0000 0FFFh

1. ADD AX, 1 ⇒ AX becomes '000Eh'
2. INC DH ⇒ DH becomes 'ACh'
3. PUSH EAX

Stack:

0E
00
00
00

ESP = 0000 0FFBh

4. PUSH ECX ⇒ Stack:

0E
00
00
00
DD
00
00
00

ESP = 0000 0FF7h

5. CMP CH, 0 ⇒ CH = 00h
6. JNZ L1 ⇒ doesnot jump
7. PUSH ECX

Stack:

0E
00
00
00
DD
00
00
00
DD
00
00
00

ESP = 0000 0FF3h

23k-2001

Date_____ 20___

8. POP EBX ⇒ EBX becomes '0000 00DDh'

Stack:

0E
00
00
00
DD
00
00
00

ESP = 0000 0FF7h

9. L1: NOT DL ⇒ DL becomes '32h'

10. PUSH EDX

(EDX = 0000 AC32h)

Stack:

0E
00
00
00
DD
00
00
00
32
AC
00
00

ESP = 0000 0FF3h

11. POP EAX ⇒ EAX becomes '0000 AC32h'

Stack:

0E
00
00
00
DD
00
00
00

ESP = 0000 0FF7h

Final Values

EAX = 0000 AC32h

EBX = 0000 00DDh

ECX = 0000 00DDh

EDX = 0000 AC32h

Ans.

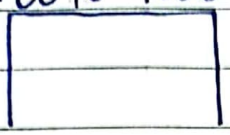
23k-2001

Date ____ 20__

b.

1. CALL Clrscr (Assuming ESP = 0000 OFFFh)

EIP = 0040 1000



→

EIP = 0040 1023

05
10
40
00

ESP = 0000 OFFB

2. RET from Clrscr PROC

EIP = 0040 1028

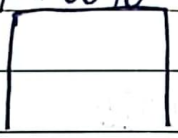
05
10
40
00

→

EIP = 0040 1005

3. CALL ArraySum

EIP = 0040 1014



→

EIP = 0040 1029

19
10
40
00

ESP = 0000 OFFB

4. Instructions inside ArraySum PROC

EIP = 0040 1029

19
10
40
00

ESP = 0000 OFF3

→

EIP = 0040 102B

19
10
40
00
00
71
A6
00
05
00
00
00

23k-2001

Date _____ 20___

5. RET from ArraySum PROC

EIP = 0040 103B

19
10
40
00
00
71
A6
00
05
00
00
00

→

EIP = 0040 1019

6. PUSH 0

EIP = 0040 1019

→

EIP = 0040 101E

00
00
00
00

ESP = 0000 0FFB

23k-2001

Date ____ 20__

Q2:

a.

```
INCLUDE Irvine32.inc
.data
  ArrList byte "computer organization and asembly language", 0
  toReplace byte ?
  msgfound byte "Here is the updated array or string", 0
  msgNfound bytes "Sorry, character not found in array or string", 0
.code
  main PROC
    mov edx, OFFSET ArrList
    call WriteString
    call Crlf
    mov eax, 0
    call ReadChar
    mov toReplace, al
    mov ecx, SIZEOF Arrlist
    mov esi, OFFSET ArrList
    mov ebx, 0
  compare:
      mov al, [esi]
      cmp al, toReplace
      jne NextChar
      mov BYTE PTR [esi], '@'
      mov ebx, 1
  NextChar:
      inc esi
      Loop compare
      cmp ebx, 0
      je NotFound
      mov edx, OFFSET msgFound
      call WriteString
      call Crlf
      mov edx, OFFSET ArrList
      call WriteString
      call CrLF
      jmp done
  NotFound:
      mov edx, OFFSET msgNfound
      call WriteString
      call Crlf
  done:
      exit
  main ENDP
  END main
```

23k-2001

Date ____ 20__

b.

```
INCLUDE Irvine32.inc
.data
  myArray Dword 100 DUP(?)
  j Dword ?
.code
  main PROC
  begin:
    mov eax, j
    cmp eax, 0
    jl Finish
    cmp eax, 100
    jg Finish
    mov esi, OFFSET myArray
    mov ebx, j
    mov ecx, ebx
    inc ecx
    mov edx, [esi + ebx * 4]
    mov [esi + ecx * 4], edx
    dec j
    jmp begin
  Finish:
    exit
  main ENDP
  END main
```

23k-2001

Date ____ 20__

Q3:

a.

MOV CL, 2	al = ?	bl = ?	cl = 02h	CF = 0
MOV Al, 8Ch	al = 8Ch	bl = ?	cl = 02h	CF = 0
MOV BL, C8h	al = 8Ch	bL = C8h	cl = 02h	CF = 0
SHL Al, Cl	al = 30h	bl = C8h	cl = 02h	CF = 0
SHR BL, Cl	al = 30h	bL = 32h	cl = 02h	CF = 0
INC Cl	al = 30h	bL = 32h	cl = 03h	CF = 0
SAR BL, Cl	al = 30h	bL = 06h	cl = 03h	CF = 0
ROL Al, CL	al = 81h	b = 06h	cl = 03h	CF = 1
CLC	al = 81h	bL = 06h	cl = 03h	CF = 0
DEC CL	al = 81h	bl = 06h	cl = 02h	CF = 0
RCL Al, Cl	al = 05h	bL = 06h	cl = 02h	CF = 0
STC	al = 05h	bL = 06h	cl = 02h	CF = 1
RCR BL, CL	al = 05h	bL = 41h	cl = 02h	CF = 1
SHRD AL, BL, 2	al = 50h	bl = 41h	cl = 02h	CF = 0
SHLD BL, AL, 2	al = 50h	bl = 05h	cl = 02h	CF = 1

b.

(Assume that time is stored in DX)

```
INCLUDE Irvine32.inc
.data
    seconds WORD ?
    minutes WORD ?
    hours WORD ?
.code
    main PROC
    mov dx, 01010101010101016
    mov al, dl
    AND al, 0001 1111b
    mov seconds, ax
    mov ax, dx
    shr ax, 5
    AND ax, 11111b
    mov minutes, ax
    mov ah, dh
    shr ah, 3
    mov hours, ax
exit
main ENDP
END main
```