Question: 01

a)

Push EAX
Push ECX
Push ECX

| | | |
|---|---|---|
| | | |
| 00FF3h | 0000 00DDh | ← ESP |
| 00FF7h | 0000 00DDh | |
| 00FFFBh | 0000 000Eh | |
| 0 FFF h | | |

POP EBX = 0000 00DDh.

| | | |
|---|---|---|
| | | |
| 0FF7h | 0000 00Dh | ← ESP |
| 0FFBh | 0000 000Eh | |
| 0FFFh | | |

| | | |
|---|---|---|
| | | |
| 0FF3 | 0000 AC32h | ← ESP |
| 0FF7 | 0000 00DDh | |
| 0FF B | 0000 0000E | |
| 0FFF | | |

Pop EAX = 0000 AC32h

| | | |
|---|---|---|
| | | |
| 0FFF7h | 0000 000Dh | ← ESP |
| 0F5Bh | 0000 000C | |
| 0FFFF | | |

final value.

EAX = 0000 AC32h

EBX = 0000 000Dh

ECX = 0000 000Dh

EDX = 0000 AC32h

ESP = 0000 0FF7h.

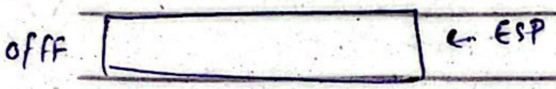
Q:1)

b)

Initially

ESP = 0000 0fffh

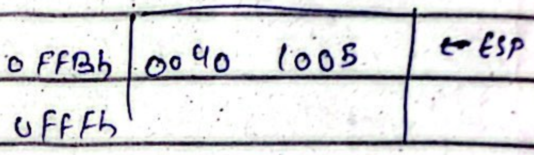
IP = 0040 100Fh



Call clrscr

ESP = 0000 0ff Bh

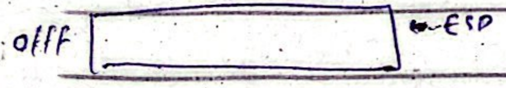
IP = 0040 1023h



RET in clrscr

ESP = 0000 0fffh

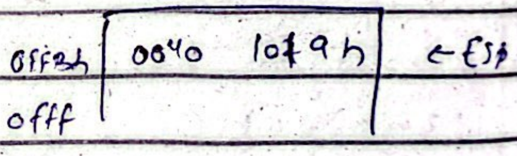
IP = 0040 1005



Call ArraySum

ESP = 0000 0fABh

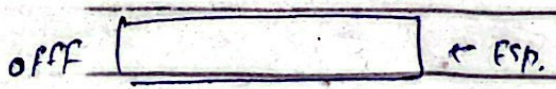
IP = 0040 102Bh



RET in ArraySum

ESP = 0000 0fffh

IP = 0040 1019h



Question (2)

a)

```
include irvine32.inc
.data
List BYTE "computer organization & assembly language",
toReplace BYTE ?
msgFound BYTE "Here is the updated array or
                    string", 0
msgNFound BYTE "Sorry, character Not found", 0

.code
main PROC

    call readchar
    move toreplace, al

     mov esi, OFFSET List
     mov ecx, LENGTHOF List
      mov ebx, 0

search:
          mov al, [esi]
          cmp al, 0        ; Null Terminator
          je check_found
          cmp al, toreplace
          jne next
          mov [esi], '@'
          inc ebx          ; flag
```

```
next:
       inc esi
    loop  search.

check-found:
         cmp ebx, 0
         je  notfound

found :
        mov  edx,  offset  msgfound
         call  writestring
         mov edx,  offset  list
         call  writestring

         jmp  end1

not_found :
             mov  edx,  offset  msgNfound
                call  writestring
  end1
     exit
     main ENDP
     END  main.
```

Qu: 2

b)

```
include irvine32.inc
.data
myarray BYTE 100 DUP(?)
j DWORD 98 ; so that myarray[j+1] will not
                        go out of bound.

.code
main PROC
   mov esi, j

  L1:
        cmp esi, 0
         je end1
        cmp esi, 98        ; cuz 99/100 +1 will go
         je  end1                          out of bound

       mov al, myarray[esi]
       mov myarray [esi+1] , al

        dec esi
        jmp L1

   end1:
   exit
   main ENDP
   END main
```

Q:3

(a)

cl = 2<br>
AL = 8ch = 1000 1100b<br>
BL = C8h = 1100 1000b

Shl AL, Cl → AL, 2<br>
: AL = 0011 0000 , CF = 0<br>
Shr BL, Cl → BL, 3<br>
: BL : 0011 0010 , CF = 0<br>
INC CL<br>
: CL = 3<br>
SAR BL, Cl<br>
: BL = 0000 0110 , CF = 0<br>
ROL AL, C<br>
: AL = 1000 0001 , CF = 1<br>
CLC<br>
: CF = 0<br>
DEC CL<br>
: CL = 2<br>
RCL AL, Cl<br>
: AL = 0000 0101 , CF = 0<br>
STC<br>
: CF = 1<br>
RCR BL, Cl<br>
: BL = 0100 0001 , CF = 0<br>
SHRD AL, BL, 2 } connot be used<br>
SHLD BL, AL, 1 } with 8 bit<br>
resistors

Q:3

(b)

```
include   irvine32.inc
.data
timestamp   DWORD ~~0~~ 01111 01111 01111 b  ; can take input too
seconds   WORD ?
minutes   WORD ?
hours     WORD ?

.code
main PROC
   ~~mov~~  ~~call~~ [illegible]
          ~~mov~~ [illegible]


    mov edx,0
    mov  edx, timestamp

    mov  eax,0
    mov  ax, dx
    add  ax,  0001 1111b
     mov  seconds , ax

      mov  eax,0
       mov  ax, dx
        shr  ax, 5
         add  ax,  0011 1111b
          mov  minutes , ax
mov eax,10
 mov al, dh
 shr al, 3
 mov hour, ax


 exit

main ENDP
 END main.
```