COAL A2 23k-0842 Sec: J

KINZA AFZAL

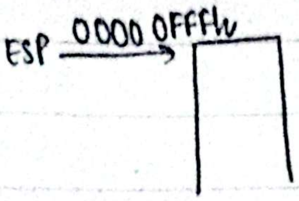
Q1 (a)

EAX = 0000 000Dh

ECX = 0000 00DDh

EDX = 0000 ~~0~~ABCDh

ESP = 0000 0FFFh



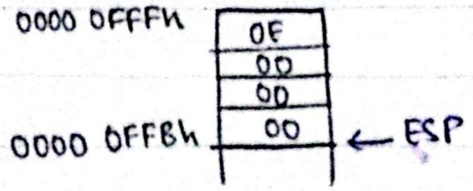
ADD AX,1

EAX = 0000 000Eh

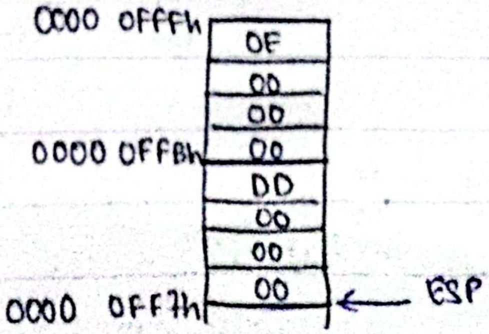
INC DH

EDX = 0000 ACCDh

PUSH EAX



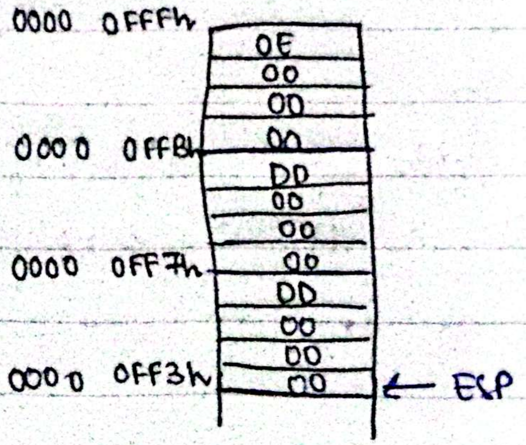
PUSH ECX



CMP CH,0 ; Equal

jnz L1 ; Since CH is 0 so this instruction is not executed

PUSH ECX



POP EBX

Pops out 0000 00DDh

0000 0FFFh	0E	
	00	
	00	
	00	
0000 0FFBh	DD	
	00	
	00	
0000 0FF7h	00	← ESP

L1:

NOT DL

EDX = 0000 AC32h

PUSH EDX

0000 0FFFh	0E	
	00	
	00	
	00	
0000 0FFBh	DD	
	00	
	00	
0000 0FF7h	00	
	32	
	AC	
	00	
0000 0FF3h	00	← ESP

POP EAX

Pops out 0000 AC32h

0000 0FFFh	0E	
	00	
	00	
	00	
0000 0FFBh	DD	
	00	
	00	
	00	← ESP

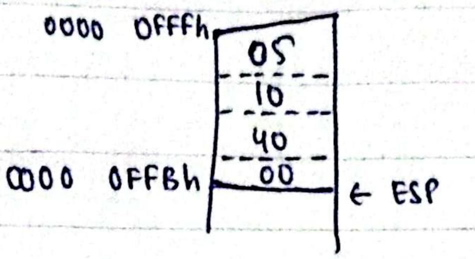
Final values :-

EAX = 0000 AC32h
EBX = 0000 00DDh
ECX = 0000 00DDh
EDX = 0000 AC32h

Q1(b)

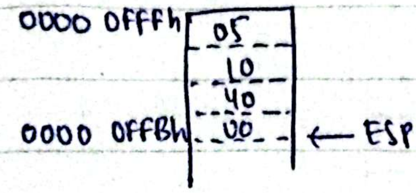
call Clrscr

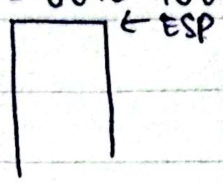
IP = 0040 1000 → IP = 0040 1023



ret (Clrscr)

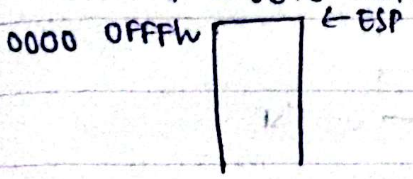
IP = 0040 1028 → IP = 0040 1005

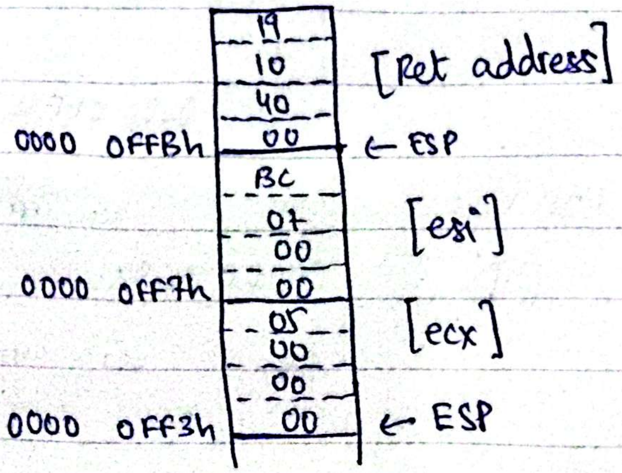




call ArraySum

IP = 0040 1014 → IP = 0040 102B

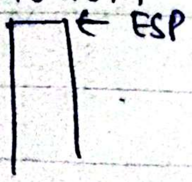




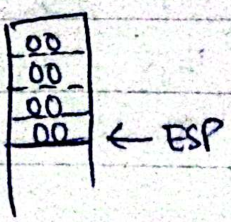
uses esi ecx

return (ArraySum)

IP = 0040 1019



push 0



Q2(a)

```
INCLUDE Irvine 32.inc .
 .data
 list BYTE "computer organization and assembly
          language",0
 toReplace BYTE ?
msgFound BYTE "Here is the updated array or string",0
msgNFound BYTE "Sorry, character not found in array
          or string",0.
flag BYTE 0
.code
main PROC
     call Readchar
     mov toReplace,al
     mov esi, OFFSET list
     mov al, toReplace
     mov bl, '@'
     mov ecx, SIZEOF list
     mov edx,0
  L1 :
      mov dl, [esi]
      cmp dl,0
      je check_found
      cmp dl, al
      jne next_char
```

```
    mov [esi], bl          ; replace with '@'
    mov [flag],1           ; flag =1

  next_char
      inc  esi    ; move to next character in string or array
  loop L1
check_found:
      cmp  [flag],1
      je   display_found
      jmp  display_not_found
display_found:
      mov  edx, OFFSET  msgFound
      call  WriteString
      call  Crlf

    mov edx, OFFSET  list
    call WriteString  ; to print modified array.
    jmp   program_end
display_not_found:
      mov  edx, OFFSET msgNFound
      call  WriteString
 program_end
    call  Crlf
    call  DumpRegs
 exit
main  ENP
END  main
```

Q2(b)

```
INCLUDE   Irvine 32.inc
 .data
   myarray   BYTE   100 DUP(0)
 .code
   main PROC
         mov    ecx , 100
   while_loop :
             cmp   ecx , 0
             jl  end_while    ; If j<0 , exit loop
             cmp   ecx , 100
             jg    end_while   ; If j>100 , exit loop
         mov   al , myarray [ecx]
         mov    myarray [ecx + 1] , al    ; myarray [j+1] = myarray [j]

         dec   ecx              ; j = j-1
         jmp   while_loop
   end_while
       call DumpRegs
       exit
   main   ENDP
   END   main
```

Q3(a)

MOV CL, 2 CL = 02h

MOV AL, 8Ch AL = 1000 1100

MOV BL, C8h BL = 1100 1000

SHL AL, CL AL = 0011 0000 (30h);

CF = 0

SHR BL, CL BL = 0011 0010 (32h).

CF = 0

INC CL CL = 03h

~~ROL AL,~~

SAR BL, CL → Arithmetic right shift

BL = 0011 0010

1st shift :-

BL = 0001 1001

2nd shift :-

BL = 0000 1100

3rd shift :-

BL = 0000 0110 (06h); CF = 0

ROL AL, CL AL = 0011 0000

first rotate :-

AL = 0110 0000 ; CF = 0

Second rotate :-

AL = 1100 0000 ; CF = 0

third rotate :-

AL = 1000 0001 ; CF = 1

(81h)

CLC Clear carry flag , CF = 0.

DEC CL ; CL = 02h

RCL AL,CL AL = 1000 0001 ; CF = 0

<u>1st rotate :-</u>

AL = 0000 0010 ; CF = 1

<u>2nd rotate :-</u>

AL = 0000 0101 ; CF = 0

STC set carry flag CF = 1.

RCR BL,CL rotate carry right

BL = 0000 0110 ; CF = 1

<u>1st rotate :-</u>

BL = 1000 0011 ; CF = 0

<u>second rotate :-</u>

BL = 0100 0001 ; CF = 1

(41h)

SHRD AL, BL, 2

destination source

AL = 0000 0101 (05h), BL = 0100 0001 (41h)

SHRD AL , BL, 2

<u>final value :-</u> AL = 0110 0000 ; CF = 0

(60 h)

SHLD BL, AL, 2

AL = 0110 0000 (60h), BL = 0100 0001 (41h)

<u>After 2 shift final value of BL :-</u>

BL = 0000 0101 (05h) , AL = 0110 0000 (60h);

CF = 1

COAL ASSIGNMENT 23K-0842, SEC : J

Q3(b) 0 to 4 bits, seconds , 5 to 10 → minutes

11 to 15 → hours , → WORD seconds, Minutes , Hours

DX register

	DH								DL							
bits →	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	0	0	1	0	0	1	1	0	0	1	1	0	1	0	1	0
	11-15 hours					5-10 minutes						0-4 seconds				

In seconds

for this (above) value of DX

Seconds in decimal are :- 10

16	8	4	2	1
0	1	0	1	0

8+2 =10 sec

Minutes :-

32	16	8	4	2	1
1	1	0	0	1	1

16+32+2+1 =51 min

Hours :-

16	8	4	2	1
0	0	1	0	0

4 hrs

∴ This is just a dry run of my assumed value, so as to verify the output.

Q3(b)

```
INCLUDE  Irvine 32.inc
.data
 Seconds  WORD ?
 Minutes  WORD ?
 Hours   WORD ?
.code
 main PROC
   mov  dx , 0010011001101010b   ; assumed value
   mov  eax,0
   mov  al,dl
   and  al, 00011111b
   mov  Seconds ,ax      ; seconds  extracted

   mov  ax , dx
   shr  ax ,5
   and  ax  , 00111111b
   mov  Minutes , ax     ; minutes  extracted

   mov  ax ,0
   mov  ah, dh.
   shr  ah, 3
   and  ah, 00011111b
   mov  Hours , ax    ; hours  extracted
call  DumpRegs
exit
main  ENDP
END  main
```