Q1_A_23k0800

Thursday, October 31, 2024 3:27 PM

Q1 (a)

Initial values:

EAX = 0000 000Dh
ECX = 0000 000Dh
EDX = 0000 ABC0h
ESP = 0000 0FFFh

add eax,1
inc dh
push eax

eax = 0000 000Eh
edx = 0000 ACC0h

ESP = 0000 0FFBh

push ecx

ESP = 0000 0FF7h

cmp ch,0
jnz L1
push ecx

zf = 0

ES = 0000 0FF3h

pop ebx

ebx = 0000 000Dh

ESP = 0000 0FF7h

Rough

C	D
1100	1101
0011	0010
3	2

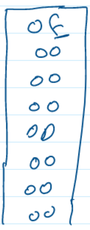
not dl
push edx

edx = 0000 AC32h

ESP = 0000 0FF3h

pop eax

eax = 0000 AC32h



ESP = 0000 0FF7h.

Ending values

EAX = 0000 AC32h
EBX = 0000 000Dh
ECX = 0000 000Dh
EDX = 0000 AC32h.

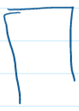
Q1_B_23k0800

Thursday, October 31, 2024 4:00 PM

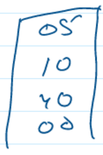
Q1 (b)

line

1 eip = 00401000



eip = 00401023

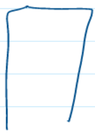


ESP = 008E FF10h

∴ From the scope of ret Class()

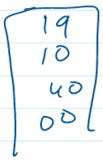
eip = 00401028

→ eip = 00401005



5. eip = 0040 1014h

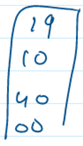
→ eip = 0040 1029



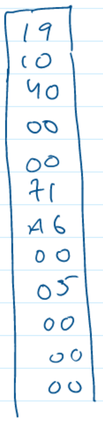
esp = 008e felon.

Uses esi, ecx

eip = 0040 1029h



→ eip = 0040 102B

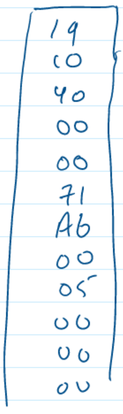


esp = 008e fe 08h

From the scope of array sum

ret

eip = 0040 103B



→ eip = 00401019

push 0

eip = 0040 1019h

→ eip = 0040 101F

esp = 008e felon

Q2 (a)

```
Include  irvine 32.inc

.data
        list  byte  "computer organization and assembly
                     language",0
        toReplace  byte  ?
        msgfound  byte  "Here is updated array",0
        msgNfound  byte  " characters not found",0


.code
  main  PROC
        mov  eax, 0
        call  readchar
        mov  toReplace, al

        mov  ecx, lengthof list
        mov  esi, offset list
        mov  ebx, 0.
    find:
        mov  al, [esi].
        cmp  al, toReplace
        je  found
        inc  esi
        loop  search
        jmp  done
  found:
        mov  [esi], '@'
        inc  ebx
        inc  esi
        loop search
  done:
        cmp  ebx, 0
        je  pfound
        mov  edx, offset  msgNfound
        call  write string
        jmp done danadone
  pfound:
        mov  edx, offset  msgfound
        call  write string
        mov  edx , offset list
        call  write string.

done danadone:
  exit
  main ENDP
END main.
```

Q2 (b)

```
include  irvine32.inc

.data
        myarray  BYTE  100  dup(?)
        j  dword  100

.code
        mov  esi , 0
        mov  esi, j
        mov  ecx, esi
    L1:
        mov  esi , ecx
        dec esi
        mov  al, onyarray [esi]
        inc  esi
        mov  myarray [esi], al.
        loop L1
exit
main  endp
end main
```

Q3

(a)

MOV CL,2	al = 00h	bl = 00h	cl = 02h	cf = 0
MOV AL,8Ch	al = 8Ch	bl = 00h	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	bl = 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

Q3b

```
include irvine32.inc
.data
timestamp   dword   0101 0101 1000 0101   (seconds)
minutes     word ?   (hours: 0101 | mins: 1000 | seconds: 0101)
seconds     word ?
hours       word ?
.code
main PROC
        mov  edx, 0
        mov  edx, timestamp
        mov  al, dl
        and  al, 00011111
        mov  seconds, ax.
        mov  ax, dx
        shr  ax, 5
        and  ax, 00111111
        mov  minutes, ax
        mov  al, dh
        and  al, 1111 1000
        shr  al, 3
        movzx  dx, al
        mov  hours, bx
exit
main endp
end main.
```