

Question 1 a)

→ considering ADD AX, 1 as line 1

Starting values:

eax = 0000 0000h

ecx = 0000 0000h

edx = 0000 ABCDh

esp = 0000 0FFFh

Line 3 :- 0000 000E ← ESP 0000 0FFBh

← Runtime stacks.

Line 4 :-	0000 000E	
	0000 00DD	← ESP 0000 0FF7h

Line 7 :-	0000 000E	
	0000 00DD	
	0000 00DD	← ESP 0000 0FF3h

Line 8 :-	0000 000E	
	0000 00DD	← ESP 0000 0FF7h

Line 10 :-	0000 000E	
	0000 00DD	
	0000 AC32	← ESP 0000 0FF3h

Ending Value

ending values

eax = 0000 AC32h

ebx = 0000 00DDh

ecx = 0000 00DDh

edx = 0000 AC32h

Line 11 :-	0000 000E	
	0000 00DD	← ESP 0000 0FF7h



Q.2 b)

1) call clrscr

eip = 00401000

eip = 00401023


00401005

← ESP 008EFE10

2) return from clrscr PROCEDURE

eip = 00401028

eip = 00401005

00401005


3) call ArraySum

eip = 00401014

eip = 00401029


00401019

← ESP 008EFE10

4) USES esi, ecx

eip = 00401029

eip = 0040102B

00401019

00401019
00A67100
00000005

← ESP 008EFE08

5) return from ArraySum PROCEDURE

eip = 0040103B

eip = 00401019

00401019
00A67100
00000005


c) push 0 :-

eip = 00401019

eip = 0040101E

0000 0000

← ESP 008EFE10



Q.2a,

:data

; same as question (given)

:code

~~mov eax, offset list~~ ~~leax, 0~~  
~~mov eax, call readchr~~

mov eax, 0

call readchr

mov to replace, al

mov esi, offset list

mov bl, '@'

mov ecx, 0

Loop:

move dl, [esi]

cmp dl, 0

je DONE

cmp dl, al

jne nextchar

mov [esi], bl

inc ecx

nextchar:

mov esi, inc esi

jmp Loop



DONE:

cmp ecx, 0

je NotFound

mov edx, offset msgFound

call writeString

call crt

mov edx, offset list

call writeString

jump skip

NotFound:

mov edx, offset msgNotFound

call writeString.

skip:

exit

; end code

END main.

Q.2 b)

loop:

mov eax, j

cmp eax, 0

jl endloop

cmp eax, 100

jg endloop

mov esi, offset myarray

mov ebx, j

mov ecx, ebx

inc ecx



mov edx, DWORD PTR [esi + ebx\*4]

mov DWORD PTR [esi + ecx\*4], edx

dec j

jmp LOOP

endloop:

Q.3a)

assumip mov C1,2 as line 1 (only changes are mentioned).

line 1 : cl = 02h

line 2 : cl = 02h, al = 8Ch

line 3 : cl = 02h, al = 8Ch, bl = 0C9h

line 4 : al = 30h, cf = 0

line 5 : bl = 32h, cf = 0

line 6 : cl = 03h

line 7 : bl = 06h, cf = 0

line 8 : al = 81h, cf = 1

line 9 : cf = 0

line 10 : cl = 2

line 11 : cf = 1

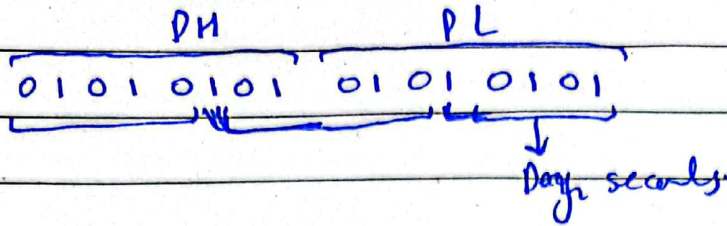
line 12 : bl = 41h, cf = 1

line 13 : bl = 41h, al = 60h, cf = 0

line 14 : bl = 05h, al = 60h, cf = 1



Q.3b)



; assuming the time is stored in dx

. data

seconds word ?

minutes word ?

hours word ?

. code

```
mov eax, 0
```

```
mov al, dl
```

```
and al, 00011111b
```

```
mov seconds, ax
```

```
mov ax, dx
```

```
shr ax, 5
```

```
and ax, 00000111 0000 11111b
```

```
mov minutes, ax
```

```
mov ax al, dh
```

```
shr al, 3
```

```
mov hours, al
```

; all three variables now have respective data.

End main