



Coal

## LAB TASK 3

NAME SAJID ISLAM

ROLLNO 24p-0745

INSTRUCTOR Muhammad Hassan

### Question 1

```
[org 0x0100]

mov al, [score]
mov bx, [population]
mov cl, [initial]
mov dl, [numbers]
mov dl, [numbers+1]
mov dl, [numbers+2]
mov dl, [numbers+3]

mov ax, 0x4c00
int 0x21

score: db 100
population: dw 2500h
initial: db 'F'
numbers: db 10,20,30,40
```

A byte variable **score** is initialized to 100 and loaded into AL (8-bit register).

A word variable **population** is set to 2500h (or 9472 decimal) and stored in BX (16-bit register).

A byte variable **initial** is assigned the character 'F' (ASCII 0x46) and moved into CL.

A 4-byte array **numbers** is defined with values 10, 20, 30, 40, each sequentially loaded into DL.

Variable	Type	Size	Value Stored	Memory Rep
Score	Byte	1	100(Decimal)	64h
population	Word	2	2500h(hex)	00h,25h,(little endian)
Initial	Byte	1	'F' (ASCII =70d)	46h
Numbers	Byte	4	10,20,30,40	0Ah, 14h,1EH,28h

## ScreenShot of DOS

DOSBox 0.74, Cpu speed: 3000 cycles, Frameskip 0, Program: AFD

Register	Value	Register	Value	Register	Value	Register	Value	Stack	Flags
AX	0064	SI	0000	CS	19F5	IP	011B	+0	0000
BX	2500	DI	0000	DS	19F5			+2	20CD
CX	0046	BP	0000	ES	19F5	HS	19F5	+4	9FFF
DX	0028	SP	FFFE	SS	19F5	FS	19F5	+6	E400

CMD >

Address	Instruction	Comment
0117 8A162701	MOV	DL,[0127]
011B B8004C	MOV	AX,4C00
011E CD21	INT	21
0120 64	DB	64
0121 0025	ADD	[DI],AH
0123 46	INC	SI
0124 0A14	OR	DL,[SI]
0126 1E	PUSH	DS
0127 28D1	SUB	CL,DL

Address	0	1	2	3	4	5	6	7
DS:0000	CD	20	FF	9F	00	EA	F0	FE
DS:0008	AD	DE	1B	05	C5	06	00	00
DS:0010	18	01	10	01	18	01	92	01
DS:0018	01	01	01	00	02	FF	FF	FF
DS:0020	FF	FF	FF	FF	FF	FF	FF	FF
DS:0028	FF	FF	FF	FF	EB	19	C0	11
DS:0030	A2	01	14	00	18	00	F5	19
DS:0038	FF	FF	FF	FF	00	00	00	00
DS:0040	05	00	00	00	00	00	00	00
DS:0048	00	00	00	00	00	00	00	00

Address	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
DS:0000	CD	20	FF	9F	00	EA	F0	FE	AD	DE	1B	05	C5	06	00	00
DS:0010	18	01	10	01	18	01	92	01	01	01	00	02	FF	FF	FF	FF
DS:0020	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	EB	19	C0	11	
DS:0030	A2	01	14	00	18	00	F5	19	FF	FF	FF	FF	00	00	00	00
DS:0040	05	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00

Step 2 ProcStep 3 Retrieve 4 Help ON 5 BRK Menu 6 7 up 8 dn 9 le 10 ri

## Question 2

```
ASM Q1.asm X ASM Q2.asm ASM Q3.asm ASM Q4.asm
ASM Q1.asm
1  [org 0x0100]
2
3  mov bl, [score]
4  mov cx, [population]
5  mov dh, [initial]
6  mov [population], ax
7  mov [score], cl
8
9  mov ax, 0x4c00
10 int 0x21
11
12 score:      db 100      ; 1 byte (decimal 100)
13 population: dw 02500h   ; 2 bytes (hexadecimal 2500h)
14 initial:    db 'F'      ; 1 byte (ASCII for 'F' = 46h)
15 numbers:    db 10 20 30 40 ; 4 bytes (array of 4 elements)
16
```

1→ Copy the value of score into the BL register.

```
DOSBox 0.74, Cpu speed: 3000 cycles, Frameskip 0, Program: AFD
AX 0000 SI 0000 CS 19F5 IP 0104 Stack +0 0000 Flags 7200
BX 0064 DI 0000 DS 19F5      +2 20CD
CX 0023 BP 0000 ES 19F5 HS 19F5 +4 9FFF OF DF IF SF ZF AF PF CF
DX 0000 SP FFFE SS 19F5 FS 19F5 +6 EA00 0 0 1 0 0 0 0 0
```

2→ Copy the value of population into the CX register.

```
CX 2500 BP 0000 ES 19F5 HS 19F5 +4 9FFF OF DF IF SF ZF AF PF CF
DX 0000 SP FFFE SS 19F5 FS 19F5 +6 EA00 0 0 1 0 0 0 0 0
```

3→ Copy the value of the initial into the DH register

```
DOSBox 0.74, Cpu speed: 3000 cycles, Frameskip 0, Program: AFD
AX 0000 SI 0000 CS 19F5 IP 010C Stack +0 0000 Flags 7200
BX 0064 DI 0000 DS 19F5      +2 20CD
CX 2500 BP 0000 ES 19F5 HS 19F5 +4 9FFF OF DF IF SF ZF AF PF CF
DX 4600 SP FFFE SS 19F5 FS 19F5 +6 EA00 0 0 1 0 0 0 0 0
```

4 → Copy the value from the AX register into the population variable.

```
DOSBox 0.74, Cpu speed: 3000 cycles, Frameskip 0, Program: AFD
AX 1234 SI 0000 CS 19F5 IP 010F Stack +0 0000 Flags 7200
BX 0064 DI 0000 DS 19F5      +2 20CD
CX 2500 BP 0000 ES 19F5 HS 19F5  +4 9FFF OF DF IF SF ZF AF PF CF
DX 4600 SP FFFE SS 19F5 FS 19F5  +6 EA00  0 0 1 0 0 0 0 0
```

5→ Copy the value from the CL register into the score variable.

```
DOSBox 0.74, Cpu speed: 3000 cycles, Frameskip 0, Program: AFD
AX 1234 SI 0000 CS 19F5 IP 0112 Stack +0 0000 Flags 7200
BX 0064 DI 0000 DS 19F5      +2 20CD
CX 2500 BP 0000 ES 19F5 HS 19F5  +4 9FFF OF DF IF SF ZF AF PF CF
DX 4600 SP FFFE SS 19F5 FS 19F5  +6 EA00  0 0 1 0 0 0 0 0
```

S or SI or SYM		64															
CMD >S																	
010F A31C01	MOV	[011C],AX															
0112 880E1B01	MOV	[011B],CL															
0116 B8004C	MOV	AX,4C00															
0119 CD21	INT	21															
011B 64	DB	64															
011C 3412	XOR	AL,12															
011E 46	INC	SI															
011F 0A14	OR	DL,[SI]															
0121 1E	PUSH	DS															

	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
DS:0000	CD	20	FF	9F	00	EA	F0	FE	AD	DE	1B	05	C5	06	00	00
DS:0010	18	01	10	01	18	01	92	01	01	01	01	00	02	FF	FF	FF
DS:0020	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	EB	19	C0	11
DS:0030	A2	01	14	00	18	00	F5	19	FF	FF	FF	FF	00	00	00	00
DS:0040	05	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00

2

= f.Ω≡■ i|..†...

.....ff. ....

δ.L.

ó.....J. ....

.....

### Question 3

```
1  [org 0x0100]
2
3  start:
4  mov al, [num]
5
6      mov ax, 0x4C00
7      int 0x21
8
9  ; --- Data Section ---
10 num: dw 1234h
11
```

We are trying to put the 16 bits data into 8 bits as al has 8 bits size while dw has 16 bits size so this is a size mismatch error

DOSBox 0.74, Cpu speed: 3000 cycles, Frameskip 0, Program: AFD

AX	0034	SI	0000	CS	19F5	IP	0103	Stack	+0	0000	Flags	7200
BX	0000	DI	0000	DS	19F5				+2	20CD		
CX	000A	BP	0000	ES	19F5	HS	19F5		+4	9FFF	OF	DF
DX	0000	SP	FFFE	SS	19F5	FS	19F5		+6	EA00	0	0

Half of value has copied to register but full value is not there as shown

## Corrected code

```
1  [org 0x0100]
2
3  start:
4      mov ax, [num]
5
6
7      mov ax, 0x4C00
8      int 0x21
9
10
11  num: dw 1234h
12
```

It will not give us logical errors because sizes of both are compatible with each other.

DOSBox 0.74, Cpu speed: 3000 cycles, Frameskip 0, Program: AFD									
AX	1234	SI	0000	CS	19F5	IP	0103	Stack	+0 0000
BX	0000	DI	0000	DS	19F5			Flags	7200
CX	000A	BP	0000	ES	19F5	HS	19F5	+2	20CD
DX	0000	SP	FFFE	SS	19F5	FS	19F5	+4	9FFF
								OF	DF
								IF	SF
								ZF	AF
								+6	EA00
								0	0
								1	0
								0	0

The whole value has been copied to register Ax

---

## Question 4

```
1  mov [var1], 99h      ; mov ax, 0099h
2
3  mov ax, [var2]        ; [correct code] mov al, [var2] load single byte into AL
4  mov bl, var1          ; [correct code] mov bx,[var1] load the 1 byte into the bl
5
6
7  var1 dw 1234h
8  var2 db 12h
```

In the first line we have added the 8 bits to al while the error was that we were assigning the 1 byte data to 2 byte which was causing mismatch error

In the second line the error was also the mismatch we were assigning the the 16 bits data to 8 bits of register which we have fixed and save the data 16 bits