

①

## ADDRESSING MODES

THERE ARE VARIOUS WAYS OF GETTING AT AN OPERANDS VALUE. THESE ARE CALLED ADDRESSING MODES.

FOR EXAMPLE THE FOLLOWING INSTRUCTION

ADD CX, [BX]

↑            ↑            ↑  
OPERATION    DESTINATION    SOURCE  
OPERAND      OPERAND      OPERAND

(REGISTER ADDRESSING)      (REGISTER INDIRECT ADDRESSING)

THE ABOVE INSTRUCTION ADDS THE CONTENTS OF SOME MEMORY LOCATION WHOSE ADDRESS IS IN BX. THE RESULT IS STORED IN CX.

(2)

# DEMI-ASM

NOTE: USE OF DW TO DECLARE A MEMORY VARIABLE.

E.G.

.DATA				INITIAL VALUE	
	MYVAR1	<u>DB</u>	0	;	ALLOCATE <u>1</u> <u>BYTE</u> IN DATA SEGMENT
	MYVAR2	<u>DW</u>	0	;	ALLOCATE A <u>WORD</u> - <u>TWO</u> BYTES
	MYVAR3	<u>DD</u>	0	;	ALLOCATE <u>DOUBLE WORD</u> ( <u>4 BYTES</u> )

16 BITS    1 BYTE (8 BITS)

MOV    AX, MYVAR1    X

16 BITS    2 BYTES (16 BITS)

MOV    AX, MYVAR2    ✓

REGISTER    MEMORY LOCATION

MOV    MYVAR2, AX

RAM    MEMORY LOCATION    REGISTER

③

title deml.asm

.model small

.stack 100h

.data

① myvar1 dw 0

② myvar2 dw 0

.code

③ ① mov ax, @data ; address of

② mov ds, ax ; data segment  
; into ax

③ mov cx, 5

④ mov myvar2, 4 ↗ TREATED AS  
DECIMAL NO.

⑤ mov bx, offset myvar2

⑥ add cx, [bx] ; pointer to memory

⑦ mov ax, 4C00h

⑧ int 21h

end

④

RUN PROGRAM IN DEBUGGER:

TASM DEM1

PRESS **F7**

TLINK DEM1

+

TO STEP THROUGH

TD DEM1

EACH INSTRUCTION

LET 1ST TWO LINES OF CODE RUN

BY PRESSING **F7**

EXAMINE DATA SEGMENT BY CLICKING  
VIEW, AND THEN DUMP.

YOU WILL SEE:

0000 MYVAR1 MYVAR2  
DS: 

00	00	00	00	00	00	00	00
----	----	----	----	----	----	----	----

  
BYTE BYTE BYTE BYTE

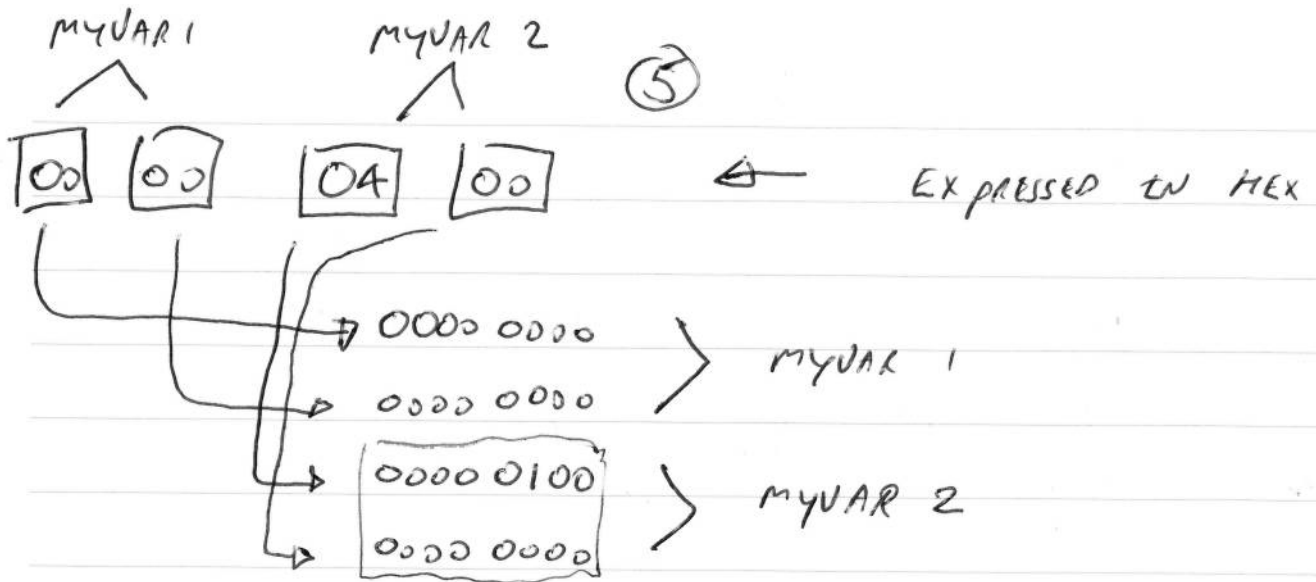
AFTER LINE ④ IS EXECUTED WHAT WILL  
THE DATA SEGMENT LOOK LIKE ?

(IF WE USE THE DEBUGGER WE WILL SEE  
THIS)

EXPRESSED IN HEXADECIMAL.  
DS 0000: 

MYVAR 1		MYVAR 2	
00	00	04	00

4 decimal = 4 hex



Change line ④ of the program to become

```
④            MOV MYVAR2, 43962
```

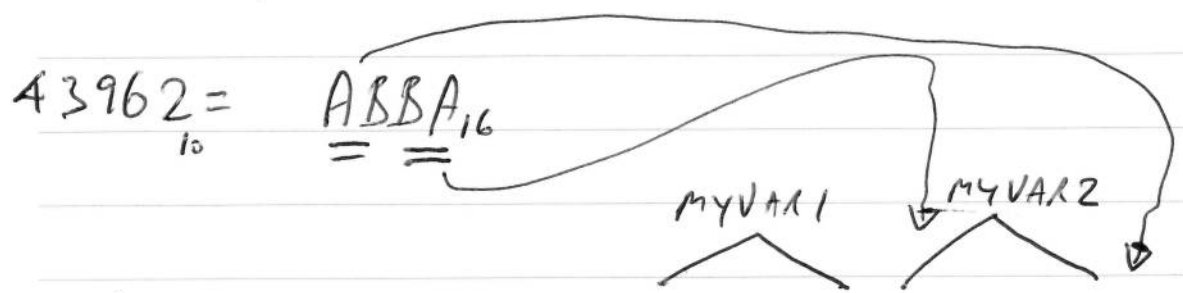
EXER.

CONVERT 43962 TO HEXADECIMAL  
 AND DETERMINE WHAT VALUES YOU WILL SEE  
 IN THE DATA SEGMENT.

(6)

16	43962			
16	2747	R	0	10 .. A
16	171	R	0	11 .. B
16	10	R	0	11 .. B
	0	R	10	10 .. A

CONVERT ~~D~~ DECIMAL TO HEX (AS ABOVE)  
OR (CONVERT TO BINARY AND CONVERT  
BINARY NO. TO HEX MAY BE EASIER)



DATA SEGMENT: 00 00 BA AB  
CONTAINS

PUT IN THE FOLLOWING LINE IN CODE

```
(4A) MOV AX, MYVAR2
```

WHAT VALUE WILL BE IN AX? AFTER (4A)  
(USE DEBUGGER TO FIND OUT)

AX: ABBA

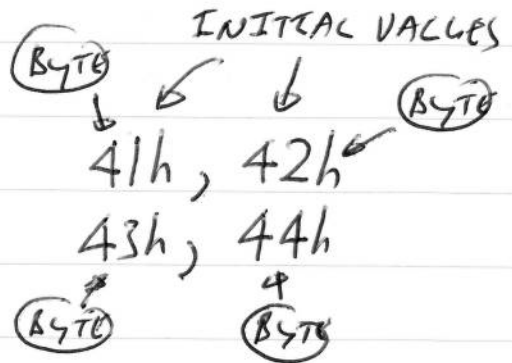
7

EXER

CHANGE DATA SEGMENT TO  
THE FOLLOWING

DATA

MYVAR1 DW  
MYVAR2 DW



WHEN SEEN IN THE DEBUGGER, YOU WILL SEE

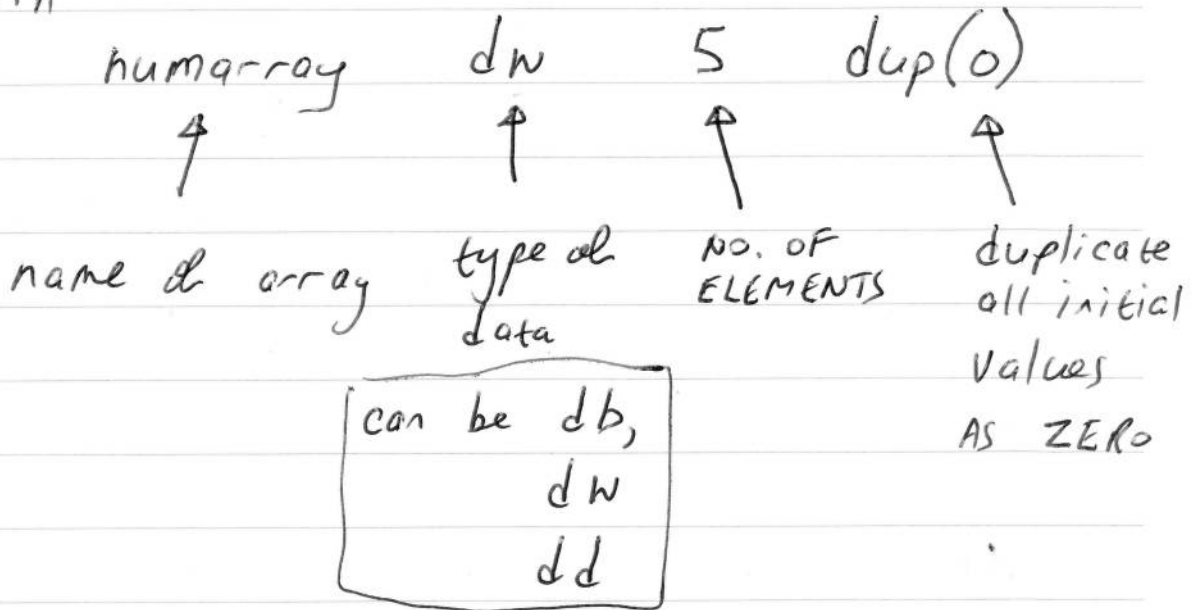
DS:0000 41 00 42 00 43 00 44 00 A B C  
ASCII VALUE

WHAT ABOUT ARRAYS? SEE LEC1198.DOC

(8)

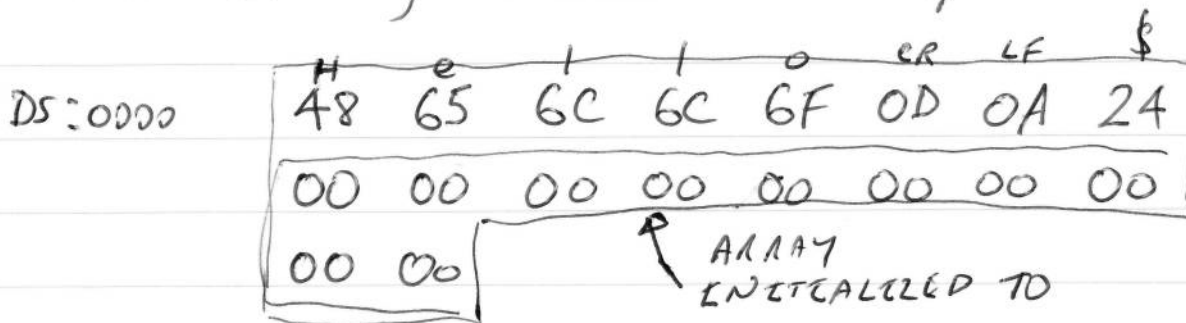
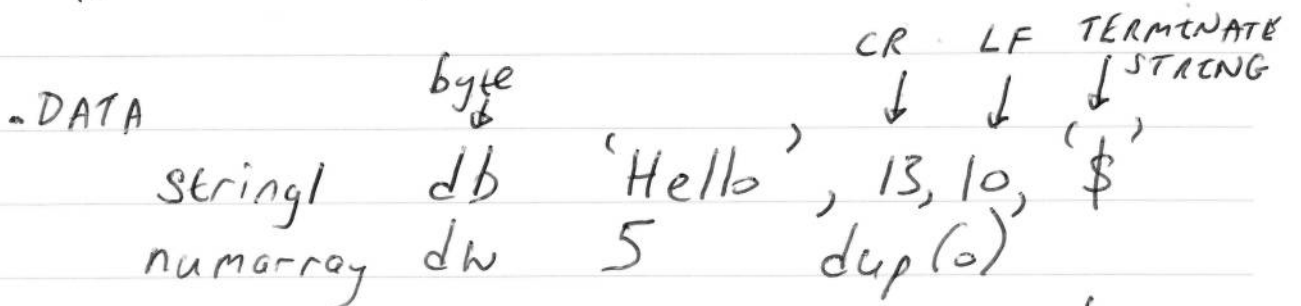
## BLOCK OF STORAGE

• DATA



WHAT WILL THE ABOVE ARRAY LOOK LIKE ?

SAY, FOR EXAMPLE WE ALSO DECLARE A STRING.



↑ ARRAY INITIALIZED TO ZERO

=