Course Title: Microprocessor and Interfacing Sessional (CSE-3812)

Department of Computer Science and Engineering (CSE)
Dhaka University of Engineering & Technology (DUET), Gazipur

Lab # 06

Understanding Advanced 8086 I/O Instructions using Array and String in Assembly Language Program.

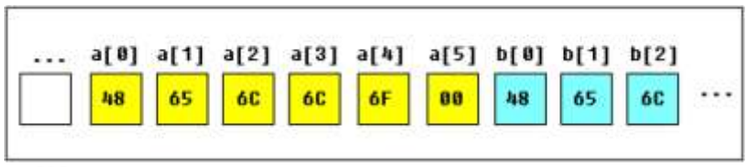
Theory:

Array:

Arrays can be seen as chains of variables. A text string is an example of a byte array; each character is presented as an ASCII code value (0..255). Some array definition examples are as follows:

a DB 48h, 65h, 6Ch, 6Ch, 6Fh, 00h
b DB 'Hello', 0

b is an exact copy of the a array, when compiler sees a string inside quotes it automatically converts it to set of bytes. This chart shows a part of the memory where these arrays are declared:
You can access the value of any element in an array using square brackets, for example:



MOV AL, a[3]

You can also use any of the memory index registers **BX, SI, DI, BP**, for example:

MOV SI, 3
MOV AL, a[SI]

If you need to declare a large array with same value you can use **DUP** operator. The syntax for **DUP**: For example:

c DB 5 DUP(0)
c DB 0, 0, 0, 0, 0 ; is an alternative way of

declaring one more example:

d DB 5 DUP(1, 2)
d DB 1, 2, 1, 2, 1, 2, 1, 2, 1, 2 ; is an alternative way of declaring

one more example:

line DB 5, 4, 3 DUP (2, 3 DUP (0) , 1)

which is equivalent to

line DB 5,4,2,0,0,0,1,2,0,0,0,l,2,0,0,0,l

Of course, you can use **DW** instead of **DB** if it's required to keep values larger then 255, or smaller then -128. **DW** cannot be used to declare strings!

String:

- **String Display Instruction**

At first define the string to be displayed under DATA SEGMENT:

.DATA
test_string DB 'My first string', 0Dh, 0Ah, '$'

Then, display the string in the command prompt as:

MOV AH, 9

LEA DX, test_string

INT 21h

Assembly Language Program Example 1 for Array:

Replace each lowercase letter in the following string by its upper case equivalent.

msg DB 'this is a message'

```
org 100h
.DATA ; Data segment starts

message db 'this is a message:$'  ;1-D array for
string
.CODE ; Code segment starts
MAIN PROC
mov ax, @DATA
mov ds, ax

lea dx, message
mov ah, 09h ;display string function
int 21h ;display message

mov ah, 02h
mov dl, 0Dh
int 21h
mov dl, 0Ah
int 21h

mov cx, 17
xor si, si
```

```
TOP:
cmp message[si],' '
je next
and message[si], 0dfh


next:
inc si
loop top

lea dx, message
mov ah, 09h ;display string function
int 21h ;display message

mov ah, 4ch
int 21h
MAIN ENDP
END MAIN
RET
```

Assembly Language Program Example 2 for Array:

To find summation of a series 1 + 2 + 3 + ... + N using array. Here, value of N is given by user where N=3 and output is shown in the output window:

```
org 100h
.DATA  ; Data segment starts
A db 3, 1, 2  ;1-D array for number B db 00h
message db 'Enter the value of N:$' ;1-D array for string
.CODE  ; Code segment starts
MAIN PROC
   mov ax, @DATA
   mov ds, ax
   xor ax,ax
   mov si, OFFSET A
   mov di, OFFSET B
   mov dx, OFFSET message ; Load Effective Address of the message in DX register
   ; lea dx, message ; (similar meaning like Load Effective Address)
   mov ah, 09h                                    ;display string function
   int 21h                                        ;display message
   mov ah, 01h
   int 21h
   mov cl, al
   sub cl, 48              ; to convert the ascii value of 3 to decimal
   3 xor al, al
   Loop_1:
         add al, [si]
         inc si
         loop Loop_1
  mov bl, al
  add bl, 48              ; to convert the ascii value of the output to decimal
  mov ah, 02h
  mov dl, 0Dh
  int 21h
```

```
mov dl, 0Ah
int 21h
mov dl, bl
int 21h
mov ah, 4ch
int 21h
MAIN ENDP
END MAIN
RET
```

Tasks to do:

1. Write an assembly language program that stores a string in a variable. Now, first display the whole string and then display the first small letter and last small letter in the string. If no small letters are entered, then display "No small letters".

Sample Input / Output:

Input in a String: input_string DB 'WE aRE DUET STuDeNTs', 0Dh, 0Ah, '$'
Output: a
s

2. Write an assembly language code to derive the final value of the number sequence $1^2+2^2+3^2+4^2+\ldots..+N^2$. (**use ARRAY and Loop**). Take the input value of N (in between 2 to 9) as a single ASCII character and then adjust it to actual decimal value in your program. Finally, store and show the output in a variable named RESULT.

Sample Input / Output:

Input: The value of N in between 2 ~9
The result is: 285

3. Write an assembly code to sort the following data in ascending order using any sorting algorithm.

Sample Input:	Sample Output:
2 6 1 9 4	The sorted list is: 1 2 4 6 9