# Lab Sheet 13 – Array Addressing

## Objectives

The objective of this lab is to practice how to use different array addressing modes.

**Tasks**

**Sample Program:** The program finds sum of three 8-bit values and places it in memory at location SUMS. Then compute sum of three word variables, and place it in memory at location SUMS+2. Data used is given below:

BYTE\_ARRAY DB 10h, 20h, 30h

WORD\_ARRAY DW 1000h, 2000h, 3000h

SUMS DW 0, 0

|  |
| --- |
| ORG 100H  .DATA  BYTE\_ARRAY DB 10H,20H,30H  WORD\_ARRAY DW 1000H,2000H,3000H  SUMS DW ?  .CODE  MOV CX,3  MOV BX,OFFSET BYTE\_ARRAY  FOR:  ADD AL,[BX]  INC BX  LOOP FOR    MOV SUMS,AX    XOR AX,AX  MOV CX,3  MOV BX,OFFSET WORD\_ARRAY  TOP:  ADD AX,[BX]  ADD BX,2  LOOP TOP    MOV [SUMS+2],AX    RET |

## Exercise

**TASK 1:** A program that counts the number of characters in a $ terminated string stored in memory:

*For Example:* for a string str1:

str1 db 'abc de', $

Total number of characters: 6 ]

\

**SOLUTION CODE:**

org 100h

.data

str1 db 'abc def$'

counter db 0

.code

MOV AX,@DATA

MOV DS,AX

lea si, str1

up:

cmp [si], '$'

je terminate

inc counter

inc si

jmp up

terminate:

mov ah, 2

add counter, 30h

mov dl, counter

int 21h

ret

**TASK 2:** Write a program to reverse a stored string of 10 characters (without using stack).

**SOLUTION CODE:**

org 100h

.data

tab db '123456789' ,13,15,'$'

.code

main proc

mov di,offset tab+10

mov cx,11

E1:

mov dl,[di]

mov ah, 2

int 21h

dec di

loop E1

main endp

end main

**TASK 3:** Practice Section 10.6 of your book.

**SOLUTION CODE:**

.MODEL SMALL

.STACK 100H

.DATA

FIVE DW 5

SCORES DW 67,45,98,33 ;Mary Allen

DW 70,56,87,44 ;Scott Baylis

DW 82,72,89,40 ;Georqe Frank\_

DW 80,67,95,50 ;Beth Harris

DW 78,76,92,60 ;Sam Wong

AVG DW 5 DUP (0)

MAIN PROC

MOV AX, @DATA

MOV DS,AX

MOV SI,6

REPEAT:

MOV CX,5

XOR BX,BX

XOR AX,AX

FOR:

ADD AX,SCORES[BX+SI]

ADD BX,8

LOOP FOR

XOR DX, DX

DIV FIVE

MOV AVG[SI],AX

SUB SI, 2

JNL REPEAT

MOV AH, 4CH

INT 21H

MAIN ENDP

END MAIN

:- o XOR DX, DX . ;clear high part of divnd

DIV. FIVE. ;AX - averaqe

HOV A"/G[SIJ,AX\_ ;

store in .:array

SUB SI, 2 - ;90 to next column

; until j=O

,JNL ·- REPEAT o;unless SI < 0 o

.34:o;dos· exit

35: MOV AH, 4CH

36: INT 21 H \_-\_\_

37: MAIN ENDP

38: END MAIN