# Lab Week 14 – Array Addressing

## Objectives

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

## Lab 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

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

|  |
| --- |
| org 100h  .DATA  str1 db 'abc d','$'  output db 'Total number of characters: $'  .CODE  MOV AX , @DATA  MOV DS , AX  MOV ES , AX  lea si,str1  mov cl,0  CLD ; Clear DF  for:  mov ax,[si]  cmp al,24h ;compare $  je exit  inc cl  movsb  jmp for  exit:  lea dx,output  mov ah,9  int 21h  mov ah,2  add cl,48d  mov dl,cl  int 21h |

|  |
| --- |
| .DATA  str db 'Asad$'  .CODE  lea si,str  mov cx,0  mov bx,si  loops:  cmp [si],'$'  je exit  inc cx  inc si  jmp loops  exit:  add cx,bx  dec bx ;to access till last element  dec cx ;to remove $  lbl:  cmp cx,bx  je term  mov ah,2  mov si,cx  mov dl,[si]  int 21h  dec cx  jmp lbl  term:  mov ah,4ch  int 21h |