-- Report--

CO215, CO Lab 2021, Assignment 5

Roll number: CSB19057

Name: PRASANJIT DUTTA

Objectives:

1. To get acquainted with reading integer input from user
2. To understand and use indirect addressing mode
3. To become familiar with loops and transfer of control instructions

Exercises:

1. List the different addressing modes used in the program (and point out where)

Instruction	**Source Address Mode**	**Destination Address Mode**
mov cx, MAX_SIZE	Direct addressing mode	Register addressing mode
mov **ah, Olh**	Immediate address mode	Register addressing mode
mov bx, OFFSET array	Register addressing mode	Register addressing mode
lea dx, prompt_msg	Direct addressing mode	Register addressing mode
sub ax, ax	Register addressing mode	Register addressing mode
sub dx, dx	Register addressing mode	Register addressing mode
mov ah, 09h	**Immediate** address mode	Register addressing mode
sub al, 30h	Immediate address mode	Register addressing mode
mov ah, 0	Immediate address mode	Register addressing mode
mov [bx]. ax	Register addressing mode	Register Indirect addressing mode
add bx, 2	Immediate address mode	Register addressing mode
mov bx. OFFSET array	Register addressing mode	Register addressing mode

mov cx, MAX_SIZE	Direct addressing mode	Register addressing mode
mov ax. 0	Immediate address mode	Register addressing mode
add ax, [bx]	Register Indirect addressing mode	Register addressing mode
add bx, 2	Immediate address mode	Register addressing mode
Mov al,0	Immediate address mode	Register addressing mode
mov sum, ax	Register addressing mode	Direct addressing mode
lea dx, output_msg	Direct addressing mode	Register addressing mode
div ten	Immediate address mode	Register addressing mode
sub dx, dx	Register addressing mode	Register addressing mode
mov ax, sum	Direct addressing mode	Register addressing mode
mov ah, 09h	Immediate address mode	Register addressing mode
mov bx, ax	Register addressing mode	Register addressing mode
add bl, 30h	Immediate address mode	Register addressing mode
add bh. 30h	Immediate address mode	Register addressing mode
mov dl, bl	Register addressing mode	Register addressing mode
mov ah, 02h	Immediate address mode	Register addressing mode
mov ah ,4ch	Register addressing mode	Register addressing mode
mov ah, 02h	Immediate address mode	Register addressing mode
mov dl, bh	Immediate address mode	Register addressing mode

2. **Rewrite the program to allow reading any number of integers terminated by 0(up to a maximum of 20)**

```
CR EQU 0DH
 LF EQU 0AH
 MAX_SIZE EQU 20
 .MODEL SMALL
 .STACK 100H
 .DATA
 array DW MAX_SIZE DUP(?)
 sum DW 0
 prompt_msg DB 'Enter 20 non-zero positive integers less than 10', CR,LF, CR, LF, '$'
 output_msg DB 'Sum of integers are: $'
 ten DB 10
 .CODE
 main PROC
  .STARTUP
  lea dx, prompt_msg
  mov ah, 09h
  int 21h
  mov bx, OFFSET array
mov cx, MAX_SIZE
sub dx, dx
rd_loop: ;reading a set of 5 single digit numbers
sub ax, ax
mov ah, 01h ; read character from  keyboard
int 21h;1 byte char gets stored in AL
sub al, 30h ; convert from character to number
mov ah, 0; AH = 0, AL = [num] : AX = [num]
cmp al,0
je compute_sum
mov [bx), ax
add bx, 2
loop rd_loop
compute_sum:
mov bx, OFFSET array
mov cx, MAX_SIZE
mov ax, 0
sum_loop:
add ax, [bx]
add bx, 2
loop sum_loop
mov sum, ax
lea dx, output_msg
mov ah, 09h
int 21h ; print a two digit integer value
sub dx, dx
mov ax, sum
div ten; al = ax/10, ah = ax%10
mov bx, ax ; keeping a copy in bx because ax will be used for display operations
```

```
add bh, 30h ; converting number to character
add bl, 30h; converting number to character
mov dl, bl
mov ah, 02h
int 21h
mov dl, bh
mov ah, 02h
int 21h
.EXIT
mov ah ,4ch
mov al, 0
int 21h
main ENDP
  END main
```

3. **Rewrite the program to allow reading two digit integers**

```
CR EQU 0DH
LF EQU 0AH
MAX_SIZE EQU 5
.MODEL SMALL
.STACK 100H
.DATA
array DW MAX_SIZE DUP(?)
sum DW 0
dig DB 0
prompt_msg DB 'Enter 5 single digit positive integers', CR, LF, CR,LF, '$'  ;
output_msg DB 'Sum of  integers are : $';
ten DB 10
.CODE
main PROC
.STARTUP
lea dx, prompt_msg
mov ah, 09h
int 21h
mov bx, OFFSET array
mov cx, MAX_SIZE
sub dx, dx
rd_loop:  ; for reading a set of double digit numbers
sub ax, ax
mov ah, 01h ; read character from the keyboard
int 21h
sub al, 30h ; from character to number
mov ah, 0; AH = O, AL = [num] : AX = [num]
cmp al,0
je compute_sum
mul ten
mov dig,al
sub ax,ax
mov ah,01h
```

```
int 21h
add al,dig
sub al, 30h ; convert from character to number
mov ah, 0; AH = 0, AL = [num] : AX = [num]
cmp al,0
mov [bx],ax
add bx, 2
loop rd_loop
compute_sum:
mov bx, OFFSET array
mov cx, MAX_SIZE
mov ax, 0
sum_loop:
add ax, [bx]
add bx, 2
loop sum_loop
mov sum, ax ; display output message
lea dx, output_msg
mov ah, 09h
int 21h ; print a two digit number sum
sub dx, dx
mov ax, sum
div ten ; al = ax/ 10, ah = ax%10
mov bx, ax ;
add bh, 30h ; convert number to character
add bl, 30h ; convert number to character
mov dl, bl
mov ah, 02h
int 21h
mov dl, bh
mov ah, 02h
int 21h
.EXIT
mov ah ,4ch
mov al,0
int 21h
main ENDP
END main
```

Learning outcome:

The exercises has led to understanding of way of reading integer input from the user. It also led to the understanding of the use of indirect addressing mode and made us familiar with loops and transfer of control instructions