**Assignment No : 9**

**Title:** Write X86/64 ALP to perform overlapped block transfer with string specific instructions Block containing data can be defined in the data segment.

**Program :**

global \_start \_start: section .text %macro accept 2 mov rax,0 mov rdi,0 mov rsi,%1 mov rdx,%2

syscall

%endmacro %macro disp 2

mov rax,1 mov rdi,1 mov rsi,%1 mov rdx,%2

syscall

%endmacro menu\_label: disp menu,menulen disp cho,lenc accept choice,02 mov al,byte[choice] cmp al,31h je successive\_add cmp al,32h je add\_shift cmp al,33h

je exit

successive\_add: disp msg1,len1 accept num,03 call convert

mov [no1],al ;storing converted first no in variable disp msg2,len2 accept num,03 call convert cmp al,00 ; multiplier=0 EX. 3\*0 je m2 ;00 present in al i.e.lower bits of ax mov [no2],al mov bx,0000h mov [result],bx mov bx,[no1] m1:

add [result],bx dec byte[no2] jnz m1 disp res,lres ;Result: m2: mov ax,[result] call display disp msg,len ;new line jmp menu\_label add\_shift: disp msg1,len1 accept num,03 call convert mov [no1],al disp msg2,len2 accept num,03 call convert mov [no2],al disp res,lres

mov bx,0000h

mov [res],bx mov ax,[no1] mov bx,[no2] as3: shr bx,01 jnc as1 add [res],ax as1: shl ax,01 cmp ax,00

jz as2 cmp bx,00 jnz as3 as2: mov ax,[res] call display disp msg,len jmp menu\_label exit:

mov rax,60 mov rdi,0 syscall display mov rsi, disparr+03 mov rcx,04 l4:

mov rdx,0 mov rbx,10h div rbx cmp dl,09h jbe add30 add dl,07h add30: add dl,30h mov [rsi],dl dec rsi dec rcx jnz l4

mov rax,1 mov rdx,1 mov rsi,disparr mov rdx,04

syscall ret convert: mov rsi,num mov al,[rsi] cmp al,39h

jle a1 sub al,07h a1: sub al,30h rol al,04 mov bl,al inc rsi mov al,[rsi] cmp al,39h ;to get the second number jle a2 sub al,07h a2: sub al,30h add al,bl ret section .data

menu: db "MENU for multiplication: ",10 db "1. Add and shift method",10 db "2. Exit",10 menulen: equ $-menu cho: db "Enter your choice: " lenc: equ $-cho msg: db " ",10 len: equ $-msg

msg1: db "Enter 1st number: " len1: equ $-msg1 msg2: db "Enter 2nd number: " len2: equ $-msg2 res: db "Result: " lres: equ $-res section .bss disparr resb 02 choice resb 02 num resb 03 no1 resb 02 no2 resb 02 result resb 04

**OUTPUT :**

