Assignment 2

```
;-----------------------------------------------------------------------------------------
section .data
nline db 10,10
nline_len equ $-nline
space db " "

ano db 10,"Assignment no 1B",10
   db 10,"Overlapping block transfer without string instruction: ",10
ano_len equ $-ano

pmsg db 10,"Before transfer : ",10
pmsg_len equ $-pmsg

smsg db 10,"Source Block: "
smsg_len equ $-smsg

dmsg db 10,"Destination block: "
dmsg_len equ $-dmsg

amsg db 10,"after transfer: ",10
amsg_len equ $-amsg

sblock db 11h,22h,33h,44h,55h
dblock  db 00h,00h,00h,00,00h

%macro Print 2
    mov rax, 1
    mov rdi, 1
    mov rsi, %1
    mov rdx, %2
    syscall
%endmacro
%macro exit 0
```

```
   Print nline,nline_len
   mov rax, 60
   mov rdi, 0
   syscall
%endmacro
;---------------------------------------------------------------------------
section .bss
char_ans resb 2
;---------------------------------------------------------------------------
section .text
global _start
_start:

    Print ano,ano_len
    Print pmsg, pmsg_len
    Print smsg, smsg_len
    mov rsi, sblock
    call display
    Print dmsg, dmsg_len
    mov rsi, dblock-2
    call display
    call  BT_O

        Print    amsg,amsg_len

        Print    smsg,smsg_len
        mov      rsi,sblock
        call     display

        Print    dmsg,dmsg_len
        mov      rsi,dblock-2
        call     display
exit;
;---------------------------------------------------------------------------
```

```
BT_O:
        mov     rsi,sblock+4      ;rsi point at the end of sblock 0+4=4
        mov     rdi,dblock+2      ;rdi point at th eend of dblock -2+4=2
        mov     rcx,5

back:   mov     al,[rsi]
        mov     [rdi],al

        dec     rsi
        dec     rdi

        dec     rcx
        jnz     back
RET
;-------------------------------------------------------------------------------
display:
   mov rbp, 5
next_num:
   mov al,[rsi]
   push rsi
   call disp_8
   Print space,1
   pop rsi
   inc rsi
   dec rbp
   jnz next_num
ret
;-------------------------------------------------------------------------------
disp_8:
    mov rsi, char_ans+1
    mov rcx, 2
    mov rbx, 16
next_digit:
    xor rdx,rdx
```

```
    div rbx
    cmp dl,9
    jbe add30
    add dl,07h
add30:
    add dl,30h
    mov [rsi],dl
    dec rsi
    dec rcx
    jnz next_digit
    Print char_ans,2
ret
;---------------------------------------------------------------------------------------
```

Output

```
Assignment no 1B

Overlapping block transfer without string instruction:

Before transfer :

Source Block: 11 22 33 44 55
Destination block: 44 55 00 00 00
after transfer:

Source Block: 11 22 33 11 22
Destination block: 11 22 33 44 55
```