TITLE:- ALP Program to perform non-overlap block transfer.

NAME:-   CLASS:-

BATCH:-   ROLL NO:-

```
**************************************************************************
section .data
    menumsg db 10,10,'#### Menu for Non-Overlapped Block Transfer ####',10
        db 10,'1.Block Transfer without using string instructions'
        db 10,'2.Block Transfer with using string instructions'
        db 10,'3.Exit',10
        db 10,'Enter Your Choice:',10
    menumsg_len equ $-menumsg
    wrchmsg db 10,10,'Wrong Choice Entered....Please try again!!!',10,10
    wrchmsg_len equ $-wrchmsg

    blk_bfrmsg db 10,'Block contents before transfer'
    blk_bfrmsg_len equ $-blk_bfrmsg

    blk_afrmsg db 10,'Block contents after transfer'
    blk_afrmsg_len equ $-blk_afrmsg

    srcmsg db 10,'Source block contents::'
    srcmsg_len equ $-srcmsg

    dstmsg db 10,'Destination block contents::'
    dstmsg_len equ $-dstmsg

    srcblk db 01h,02h,03h,04h,05h
    dstblk times 5 db 0
    cnt equ 05

    spacebar db 20h
    lfmsg db 10,10

section .bss
    optionbuff resb 02
    dispbuff resb 02

%macro dispmsg 2
    mov eax, 04
    mov ebx, 01
    mov ecx, %1
    mov edx, %2
    int 80h
%endmacro

%macro accept 2
    mov eax, 03
    mov ebx, 00
    mov ecx, %1
    mov edx, %2
    int 80h
%endmacro
```

```
section .text
    global _start
_start:
    dispmsg blk_bfrmsg, blk_bfrmsg_len
    call showblock

menu:
    dispmsg menumsg, menumsg_len
    accept optionbuff, 02
    cmp byte[optionbuff], '1'
    jne case2
    call blkxferwo_proc
    jmp exit1

case2:
    cmp byte[optionbuff], '2'
    jne case3
    call blkxferw_proc
    jmp exit1

case3:
    cmp byte[optionbuff], '3'
    je exit
    dispmsg wrchmsg, wrchmsg_len
    jmp menu

exit1:
    dispmsg blk_afrmsg, blk_afrmsg_len
    call showblock
    dispmsg lfmsg, 02
    jmp menu

exit:
    mov eax, 01
    mov ebx, 00
    int 0x80

blkxferw_proc:
    mov esi, srcblk
    mov edi, dstblk
    mov ecx, cnt
    cld
    rep movsb
    ret

blkxferwo_proc:
    mov esi, srcblk
    mov edi, dstblk
    mov ecx, cnt
```

```
blkup1:
    mov al, [esi]
    mov [edi], al
    inc esi
    inc edi
    loop blkup1
    ret

showblock:
    dispmsg srcmsg, srcmsg_len
    mov esi, srcblk
    call dispblk_proc

    dispmsg dstmsg, dstmsg_len
    mov esi, dstblk
    call dispblk_proc
    ret

dispblk_proc:
    mov rcx, cnt
rdisp:
    push rcx
    mov bl, [esi]
    call disp8_proc
    inc esi
    dispmsg spacebar, 01
    pop rcx

    loop rdisp
    ret

disp8_proc:
    mov ecx,02
    mov edi, dispbuff
dup1:
    rol bl, 04
    mov al, bl
    and al, 0fh
    cmp al, 09
    jbe dskip
    add al, 07
dskip:
    add al, 30h
    mov [edi], al
    inc edi
    loop dup1

    dispmsg dispbuff,02
    ret
```

```
******************************* OUTPUT **********************************

admin@oomplab:~$ nasm -f elf64 -o Pract2a.o Pract2a.nasm
admin@oomplab:~$ ld -o Pract2a Pract2a.o
admin@oomplab:~$ ./Pract2a

Block contents before transfer
Source block contents::01 02 03 04 05
Destination block contents::00 00 00 00 00

##### Menu for Non-Overlapped Block Transfer #####

1.Block Transfer without using string instructions
2.Block Transfer with using string instructions
3.Exit
Enter Your Choice:1

Block contents after transfer
Source block contents::01 02 03 04 05
Destination block contents::01 02 03 04 05

##### Menu for Non-Overlapped Block Transfer #####

1.Block Transfer without using string instructions
2.Block Transfer with using string instructions
3.Exit
Enter Your Choice:2

Block contents after transfer
Source block contents::01 02 03 04 05
Destination block contents::01 02 03 04 05

##### Menu for Non-Overlapped Block Transfer #####

1.Block Transfer without using string instructions
2.Block Transfer with using string instructions
3.Exit
Enter Your Choice:4

Wrong Choice Entered....Please try again!!!

##### Menu for Non-Overlapped Block Transfer #####

1.Block Transfer without using string instructions
2.Block Transfer with using string instructions
3.Exit
Enter Your Choice:3
admin@oomplab:~$
```