

Assignment 2

Name: Prajwal Khobragade

Roll no: SYCOB138

section .data

nline db 10,10 ; New line characters

nline_len equ \$-nline ; Length of nline

space db " "

ano db 10," Assignment no : 2-B",

db 10," Block Transfer-Non overlapped with String instruction.",

ano_len equ \$-ano

bmsg db 10,"Before Transfer::"

bmsg_len equ \$-bmsg

amsg db 10,"After Transfer::"

amsg_len equ \$-amsg

smsg db 10," Source Block : "

smsg_len equ \$-smsg

dmsg db 10," Destination Block : "

dmsg_len equ \$-dmsg

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

dblock times 5 db 0

;

section .bss

char_ans resB 2 ; Character answer buffer

;

```
%macro Print 2          ; Print macro
```

```
    MOV RAX,1
```

```
    MOV RDI,1
```

```
    MOV RSI,%1
```

```
    MOV RDX,%2
```

```
    syscall
```

```
%endmacro
```

```
%macro Read 2           ; Read macro
```

```
    MOV RAX,0
```

```
    MOV RDI,0
```

```
    MOV RSI,%1
```

```
    MOV RDX,%2
```

```
    syscall
```

```
%endmacro
```

```
%macro Exit 0           ; Exit macro
```

```
    Print nline, nline_len
```

```
    MOV RAX,60
```

```
    MOV RDI,0
```

```
    syscall
```

```
%endmacro
```

```
;------
```

```
section .text
```

```
    global _start
```

```
_start:
```

```
    Print ano, ano_len      ; Print assignment details
```

```
    Print bmsg, bmsg_len    ; Print message before transfer
```

```
    Print smsg, smsg_len    ; Print source block message
```

```
mov rsi, sblock      ; Set source block address
call disp_block      ; Call display block function
```

```
Print dmsg, dmsg_len ; Print destination block message
mov rsi, dblock      ; Set destination block address
call disp_block      ; Call display block function
```

```
call BT_NOS          ; Call block transfer function
```

```
Print amsg, amsg_len ; Print message after transfer
```

```
Print smsg, smsg_len ; Print source block message
mov rsi, sblock      ; Set source block address
call disp_block      ; Call display block function
```

```
Print dmsg, dmsg_len ; Print destination block message
mov rsi, dblock      ; Set destination block address
call disp_block      ; Call display block function
```

```
Exit                ; Exit program
```

```
;-----
```

```
BT_NOS:
```

```
mov rsi, sblock      ; Set source block address
mov rdi, dblock      ; Set destination block address
mov rcx, 5           ; Set loop counter to 5
```

```
next:
```

```
CLD                ; Clear direction flag to increment rsi, rdi
REP MOVSB          ; Repeat move byte from source to destination
                   ; (incrementing both rsi, rdi), 5 times
```

```
RET
```

```
;-----
```

```
disp_block:
```

mov rbp, 5 ; Set loop counter to 5

next_num:

mov al, [rsi] ; Move byte from source to al
push rsi ; Save current value of rsi
call Disp_8 ; Call display 8 function
Print space, 1 ; Print space character
pop rsi ; Restore value of rsi
inc rsi ; Increment source pointer
dec rbp ; Decrement loop counter
jnz next_num ; Jump to next_num if rbp is not zero

RET

;------

Disp_8:

MOV RSI, char_ans+1 ; Set rsi to point to the second byte of char_ans
MOV RCX, 2 ; Set loop counter to 2
MOV RBX, 16 ; Set base to 16 (hexadecimal)

next_digit:

XOR RDX, RDX ; Clear rdx for division
DIV RBX ; Divide al by 16, quotient in al, remainder in dl

CMP DL, 9 ; Check if remainder is less than 9
JBE add30 ; Jump if below or equal to 9
ADD DL, 07H ; Adjust remainder for letters A-F

add30:

ADD DL, 30H ; Convert remainder to ASCII character
MOV [RSI], DL ; Store character in char_ans buffer
DEC RSI ; Decrement rsi to point to the next byte
DEC RCX ; Decrement loop counter
JNZ next_digit ; Jump to next_digit if rcx is not zero

Print char_ans, 2 ; Print the content of char_ans buffer

RET

;-----

Output

```
Assignment no : 2-B
Block Transfer-Non overlapped with String instruction.
Before Transfer::
Source Block : 11 22 33 44 55
Destination Block : 00 00 00 00 00
After Transfer::
Source Block : 11 22 33 44 55
Destination Block : 11 22 33 44 55

[Execution complete with exit code 0]
```