MICROPROCESSOR LABORATORY

Assignment No. 7

NAME :- OJUS PRAVIN JAISWAL
ROLL NO. :- SACO19108
DIVISION :- A

Assignment No. 7

Program :

```
;Non Overlapped Block Transfer

%macro print 2
Mov rax,1
Mov rdi,1
Mov rsi,%1
Mov rdx,%2
syscall
%endmacro

%macro exit 0
mov rax,60
mov rdi,0
syscall
%endmacro

section .data
sblock db 10h,20h,30h,40h,50h
dblock times 5 db 0

msg1 db 10,13,"Before Non Overlapped Block Transfer :- ",10,13
msg1_len equ $-msg1
```

```
msg2 db 10,13,"After Non Overlapped Block Transfer :- ",10,13
msg2_len equ $-msg2

new_line db 10,13

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

dmsg db 10,"Destination Block is : "
dmsg_len equ $-dmsg

space db " "

section .bss
Char_ans resb 2

Section .text
global _start
_start:

print msg1,msg1_len
print smsg,smsg_len
mov rsi,sblock
call block_display

print dmsg,dmsg_len
mov rsi,dblock
call block_display
```

```
call block_transfer

print new_line,1
print msg2,msg2_len
print smsg,smsg_len
mov rsi,sblock
call block_display

print dmsg,dmsg_len
mov rsi,dblock
call block_display
print new_line,1

exit

block_transfer:
mov rsi,sblock
mov rdi,dblock
mov rcx,5
up:
mov al,[rsi]
mov [rdi],al
inc rsi
inc rdi
dec rcx
jnz up
ret
```

```
block_display:
mov rbp,5
next:
mov al,[rsi]
push rsi
call display
print space,1
pop rsi
inc rsi
dec rbp
jnz next
ret

display:
Mov rsi,Char_ans+1
mov rcx,2
mov rbx,16
up1:
xor rdx,rdx
div rbx
cmp dl,09
jbe add30
add dl,07h
add30:
add dl,30h
mov [rsi],dl
dec rsi
```

dec rcx

jnz up1

print Char_ans,2

ret

Code Input Output Run Save

```
;Non Overlapped Block Transfer

%macro print 2
Mov rax,1
Mov rdi,1
Mov rsi,%1
Mov rdx,%2
syscall
%endmacro

%macro exit 0
mov rax,60
mov rdi,0
syscall
%endmacro

section .data
sblock db 10h,20h,30h,40h,50h
dblock times 5 db 0

```

Code Input Output Run Save

```
msg1 db 10,13,"Before Non Overlapped Block Transfer :- ",10,13
msg1_len equ $-msg1

msg2 db 10,13,"After Non Overlapped Block Transfer :- ",10,13
msg2_len equ $-msg2

new_line db 10,13

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

dmsg db 10,"Destination Block is : "
dmsg_len equ $-dmsg

space db " "

section .bss
Char_ans resb 2

Section .text
```

Code Input Output Run Save

```
global _start
_start:

print msg1,msg1_len
print smsg,smsg_len
mov rsi,sblock
call block_display

print dmsg,dmsg_len
mov rsi,dblock
call block_display

call block_transfer

print new_line,1
print msg2,msg2_len
print smsg,smsg_len
mov rsi,sblock
call block_display

```

Code Input Output Run Save

```
print dmsg,dmsg_len
mov rsi,dblock
call block_display
print new_line,1

exit

block_transfer:
mov rsi,sblock
mov rdi,dblock
mov rcx,5
up:
mov al,[rsi]
mov [rdi],al
inc rsi
inc rdi
dec rcx
jnz up
ret

```

Code Input Output Run Save

```
block_display:
mov rbp,5
next:
mov al,[rsi]
push rsi
call display
print space,1
pop rsi
inc rsi
dec rbp
jnz next
ret

display:
Mov rsi,Char_ans+1
mov rcx,2
mov rbx,16
up1:
xor rdx,rdx
div rbx
```

Code Input Output Run Save

```
display:
Mov rsi,Char_ans+1
mov rcx,2
mov rbx,16
up1:
xor rdx,rdx
div rbx
cmp dl,09
jbe add30
add dl,07h
add30:
add dl,30h
mov [rsi],dl
dec rsi
dec rcx
jnz up1
print Char_ans,2
ret
```

Output :

Code Input Output Run Save

```
Before Non Overlapped Block Transfer :-

Source Block is : 10 20 30 40 50
Destination Block is : 00 00 00 00 00

After Non Overlapped Block Transfer :-

Source Block is : 10 20 30 40 50
Destination Block is : 10 20 30 40 50

[Program exited with exit code 0]
```