

```

%macro write 2
mov rax , 1
mov rdi , 1
mov rsi , %1
mov rdx , %2
syscall
%endmacro

%macro read 2
mov rax , 0
mov rdi , 0
mov rsi , %1
mov rdx , %2
syscall
%endmacro

section .data
msg1: db 0x0A,"Enter the number whose factorial you want to find :-> ",0x0A
len1: equ $ - msg1
msg2: db "The factorial is :->  "
len2: equ $ - msg2
cnt db 00h

section .bss
number resb 2
result resb 16

section .text
global _start
_start:

write msg1,len1
read number , 2

mov rsi , number
CALL AtoH          ; here the number is stored in the BX

Factorial:
CALL factProcedure
mov rbx , rax      ; the multiplication result is in the RAX to we move it to RBX
mov rdi , result   ; the pointer is added on the result value which will get the
result value after conversion
CALL HtoA
write msg2,len2
write result , 16

exit:
mov rax , 60
mov rdi , 0
syscall

factProcedure:
cmp bx,0001H
jne above1

```

```
mov ax,1
ret
above1:
push rbx;
dec bx
call factProcedure
pop rbx
mul bx
ret
```

```
AtoH:
mov word[cnt],02H
mov bx,00H
up:
rol bl,04
mov al,byte[rsi]
cmp al,39H
JBE belowNine
SUB al,07H
belowNine:
sub al,30H
add bl,al
INC rsi
DEC word[cnt]
JNZ up
ret
```

```
HtoA:
mov word[cnt],04H
up1:
rol bx,04
mov cl,bl
and cl,0FH
CMP CL,09H
jbe belowNine1
ADD cl,07H
belowNine1:
add cl, 30H
mov byte[rdi],cl
INC rdi
dec word[cnt]
JNZ up1
ret
```