%macro print 2

mov eax,4

mov ebx,1

mov ecx,%1

mov edx,%2

int 80h

%endmacro

%macro scan 2

mov eax,3

mov ebx,2

mov ecx,%1

mov edx,%2

int 80h

%endmacro

section .data

msg1 db 10,"Enter two digits of Number : ",0

msg1\_len equ $-msg1

msg2 db 10,"Multiplication of elements : ",0

msg2\_len equ $-msg2

nline db 10

section .bss

num resb 3

num1 resb 2

result resb 4

section .text

global \_start

\_start:

print msg1,msg1\_len

scan num,3

call atoh ; call ascii to hex

mov [num1],bl ; move first number to num1

print msg1,msg1\_len ; ask multiplier

scan num,3 ; scan second number

call atoh ; call ascii to hex

mov ecx,0h ; initialise ecx to 0

mov eax,[num1] ; move first number to acmltr

addup: ; successive addition procedure

add ecx,eax ; add first number in ecx

dec bl ; decrement second number

jnz addup ; jump until bl not equal to 0

mov ebx,ecx ; move result in ebx

call htoa ; call htoa

print msg2,msg2\_len

print result,4

print nline,1

mov eax,1 ; Exit

mov ebx,0

int 80h

atoh:

mov bl,0 ; initial result is 0

mov ecx,2 ; initial counter 2

mov esi,num ; copy input number in source index

up1:

rol bl,4 ; Rotate bl to 04 bit

mov al,[esi] ; move input number in lower acmltr

cmp al,39h ; compare with 39h=ascii 9

jbe skip1 ; jump if below or equals

sub al,7h

skip1:

sub al,30h ; substract 30h

add bl,al ; add remaining in bl

inc esi ; increament source index by 1

loop up1 ; loop back

ret

htoa:

mov ecx,4 ; initial counter 4

mov edi,result ; move 4 to result

dup1:

rol bx,4 ; rotate left to 4 bit

mov al,bl ; move bl to al

and al,0fh ; AND with 0fh= 0000 1111

cmp al,9 ; compare with 09

jbe S1 ; jump if below or equals

add al,7h

S1:

add al,30h ; add 30h in al

mov [edi],al ; move al in destination index

inc edi ; increament destination index pointer

loop dup1 ; loop back

ret