

```
; Don Spickler
; Calculator (factorial)
; Compile with: nasm -f elf factorial.asm
; Link with (64 bit systems require elf_i386 option): ld -m elf_i386 factorial.o -o factorial
; Or run make
; Run with: ./factorial n
```

```
%include      'functions.asm'
```

```
SECTION .data
msg1      db      '! = '      ; a message string to correctly output result
```

```
SECTION .bss
fact      resd 1
n         resd 1
```

```
SECTION .text
global _start
```

```
_start:
    pop     ecx          ; first element is the number of arguments

    mov     eax, 1
    cmp     ecx, 1        ; if ecx is 1 then halt, no argument or more than one.
    je      finish        ; jump to finish if ecx is 0

    pop     ecx          ; program name.
    pop     eax          ; the number n
    call    atoi          ; convert to numeric value
    mov     ecx, eax      ; store in ecx for calculation.

    mov     [n], ecx      ; Store n for printing later.
    mov     eax, 1        ; move 1 into eax for factorial accumulation.

    cmp     ecx, 0        ; if ecx is 0 then halt
    je      finish        ; jump to finish if ecx is 0
```

```
continue:
    mul     ecx          ; multiply eax by ecx
    dec     ecx          ; decrement ecx
    cmp     ecx, 0        ; if ecx is 0 then halt
    je      finish        ; jump to finish if ecx is 0
    jmp     continue      ; otherwise continue with the next number.
```

```
finish:
    mov     [fact], eax   ; store the result of n! in fact
    mov     eax, [n]      ; load n into eax for printing.
    call    iprint        ; call our integer printing with linefeed function
    mov     eax, msg1     ; move our message string into eax
    call    sprint        ; call our string print function
    mov     eax, [fact]   ; move answer into eax
    call    iprintLF      ; call our integer printing with linefeed function

    call    quit
```