

```
factorial:
.align 4
.section .text
.global _start

# x1: input
# x2: running output
# x3: fixed 0
# x4: factorial counter
# x5: multiplication result
# x6: multiplication counter
# x7: fixed 1

_start:
    lw x1, factorial_input # load parameter
    addi x2, x1, 0         # load parameter
    andi x3, x3, 0         # clear x3
    addi x4, x1, 0         # factorial_input
    andi x7, x7, 0         # initialize x7 for later comparison
    addi x7, x7, 1

factorial_dec_loop:
    addi x4, x4, -1        # decrement
    addi x6, x4, 0         # set mult counter to represent ex. 5*4*3*2*1...
    andi x5, x5, 0         # clear mult result every iter

multiply_dec_loop:
    add x5, x5, x2
    addi x6, x6, -1        # decrement multiplicatio counter
    bne x3, x6, multiply_dec_loop

    addi x2, x5, 0         # set x2 with new multiply result
    bne x4, x7, factorial_dec_loop # 1*result = result so continue

    la x8, answ_output     # store result
    sw x2, 0(x10)

halt:
    beq x0, x0, halt

.section .rodata
factorial_input: .word 0x00000005
answ_output: .word 0x00000000
```