

fact:

```
    addi sp, sp, -8      # adjust stack for 2 items
    sw x1, 4(sp)         # save the return address
    sw x10, 0(sp)        # save the argument n
    addi x5, x10, -1     # x5 = n - 1
    bge x5, x0, .else    # if n < 1
    addi x10, x0, 1       # return 1
    addi sp, sp, 8        # pop 2 items off stack
    jalr x0, 0(x1)       # return to caller
```

```
.else:
    addi x10, x10, -1    # n >= 1: argument gets (n - 1)
    jal x1, fact         # call fact with (n - 1)
    addi x6, x10, 0      # return from jal: move result of fact
                        # (n - 1) to x6:
    lw x10, 0(sp)        # restore argument n
    lw x1, 4(sp)         # restore the return address
    addi sp, sp, 8        # adjust stack pointer to pop 2 items
    mul x10, x10, x6      # return n * fact (n - 1)
    jalr x0, 0(x1)       # return to the caller
```