Assembly Syntax

- The x86 assembly language has two main syntax branches (same instructions but different syntax):
    - Intel syntax is dominant in the DOS and Windows world,
    - AT&T syntax is dominant in the Unix world, since Unix was created at AT&T Bell Labs.

Example:

Factorial:

```
n=5;
prod=1;
for(i=0; i<=n;i++)
        prod*=i
```

```
We will use:    %rdx for n
                %rax for prod
                %rcx for i
```

```
        mov $5, %rdx          # n=5
        mov $1, %rax          # prod=1
        mov $1, %rcx          # i=1
TOP:                          # top of the loop
        cmpq %rdx, %rcx       #compare i to n
        jq DONE               # if i>n, end the loop
        imulq %rcx,%rax       # prod*=i
        incq %rcx             # i++
        Jmp TOP
DONE:
```

Note:

- Function name is declared using a label that must start with _ in macos.
- Parameters of a function must go in order into: rdi, rsi, rdx, rcx, r8, r9 consequently.
- To run after calling the function from c:

```
gcc test.c fac.s
./a.out
```