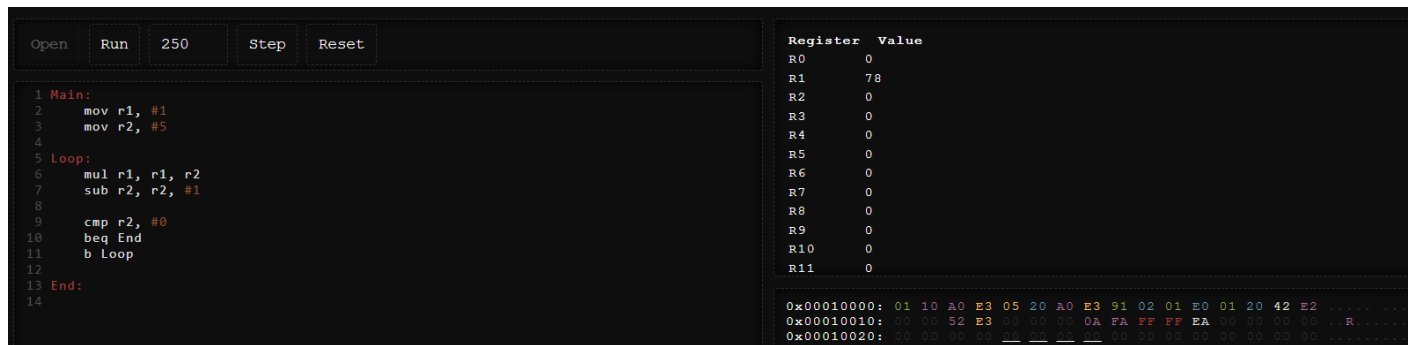


Template Week 4 – Software

Student number: 562594

Assignment 4.1: ARM assembly

Screenshot of working assembly code of factorial calculation:



The screenshot shows an ARM assembly simulator interface. On the left, there is a code editor with the following assembly code:

```
1 Main:
2   mov r1, #1
3   mov r2, #5
4
5 Loop:
6   mul r1, r1, r2
7   sub r2, r2, #1
8
9   cmp r2, #0
10  beq End
11  b Loop
12
13 End:
14
```

On the right, there is a register window showing the values of registers R0 through R11:

| Register | Value |
|----------|-------|
| R0 | 0 |
| R1 | 78 |
| R2 | 0 |
| R3 | 0 |
| R4 | 0 |
| R5 | 0 |
| R6 | 0 |
| R7 | 0 |
| R8 | 0 |
| R9 | 0 |
| R10 | 0 |
| R11 | 0 |

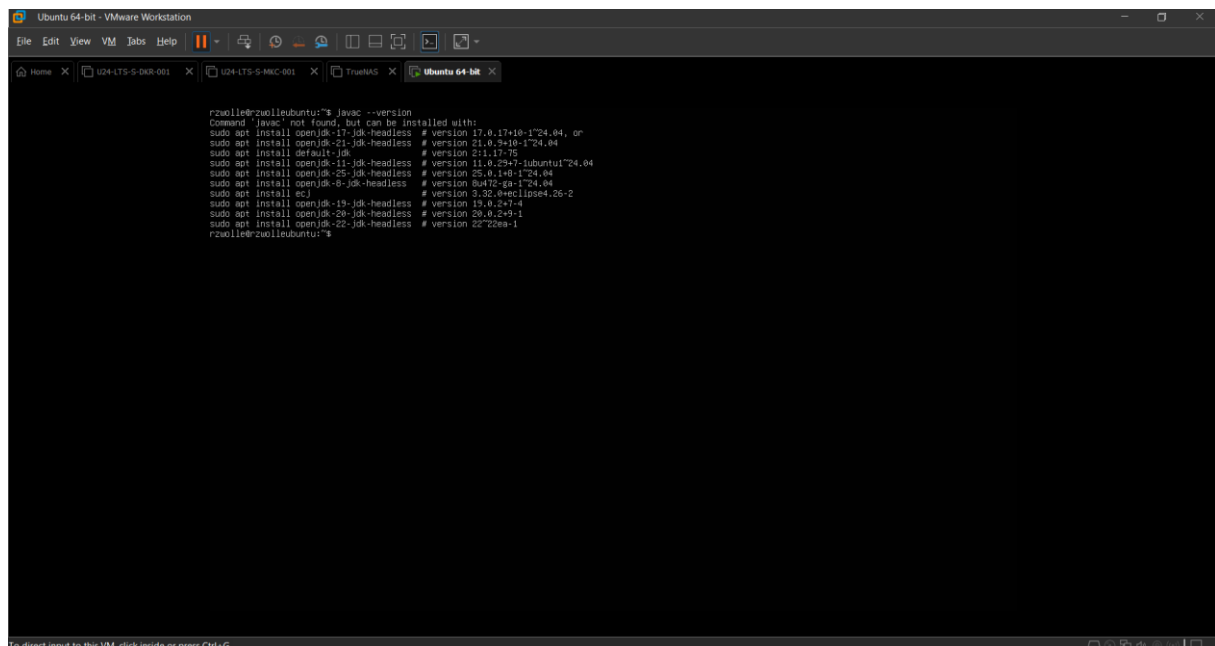
At the bottom right, there is a memory dump showing hexadecimal values and their corresponding ASCII characters:

```
0x00010000: 01 10 A0 E3 05 20 A0 E3 91 02 01 00 01 20 42 E2 ...
0x00010010: 00 00 52 E3 00 00 00 0A FA FF FF EA 00 00 00 00 ..R...
0x00010020: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 ...
```

Assignment 4.2: Programming languages

Take screenshots that the following commands work:

javac –version



The screenshot shows a terminal window in a VMware Workstation environment. The terminal output is as follows:

```
rzoolie@zuolleubuntu:~$ javac --version
Command 'javac' not found, but can be installed with:
sudo apt install openjdk-11-jdk-headless # version 17.0.17+10-1724.04, or
sudo apt install openjdk-21-jdk-headless # version 21.0.3+10-1724.04
sudo apt install default-jdk # version 21:17-75
sudo apt install openjdk-11-jdk-headless # version 11.0.29+7-1ubuntu1724.04
sudo apt install openjdk-25-jdk-headless # version 25.0.1+0-1724.04
sudo apt install openjdk-8-jdk-headless # version 8u472-b01-1724.04
sudo apt install ecj # version 3.32.0+eclipse4.26-2
sudo apt install openjdk-19-jdk-headless # version 19.0.2+7-4
sudo apt install openjdk-20-jdk-headless # version 20.0.2+0-1
sudo apt install openjdk-22-jdk-headless # version 2272ea-1
rzoolie@zuolleubuntu:~$
```

java –version

```
Ubuntu 64-bit - VMware Workstation
File Edit View VM Tabs Help
Home X U24-LTS-S-DKR-001 X U24-LTS-S-MMC-001 X TrueNAS X Ubuntu 64-bit X

rzwalie@rzwalieubuntu:~$ javac --version
Command 'javac' not found, but can be installed with:
sudo apt install openjdk-17-jdk-headless # version 17.0.17+10-1~24.04, or
sudo apt install openjdk-21-jdk-headless # version 21.0.3+10-1~24.04
sudo apt install default-jdk # version 2:1.17-75
sudo apt install openjdk-11-jdk-headless # version 11.0.29+7~ubuntu1~24.04
sudo apt install openjdk-20-jdk-headless # version 20.0.2+9-1~24.04
sudo apt install openjdk-8-jdk-headless # version 8u472-ga-1~24.04
sudo apt install ecj # version 3.32.0+eclipse4.26-2
sudo apt install openjdk-19-jdk-headless # version 19.0.2+7-1
sudo apt install openjdk-20-jdk-headless # version 20.0.2+9-1
sudo apt install openjdk-22-jdk-headless # version 22~220a-1
rzwalie@rzwalieubuntu:~$ java --version
Command 'java' not found, but can be installed with:
sudo apt install openjdk-17-jre-headless # version 17.0.17+10-1~24.04, or
sudo apt install openjdk-21-jre-headless # version 21.0.3+10-1~24.04
sudo apt install default-jre # version 2:1.17-75
sudo apt install openjdk-11-jre-headless # version 11.0.29+7~ubuntu1~24.04
sudo apt install openjdk-20-jre-headless # version 20.0.2+9-1~24.04
sudo apt install openjdk-8-jre-headless # version 8u472-ga-1~24.04
sudo apt install openjdk-19-jre-headless # version 19.0.2+7-1
sudo apt install openjdk-20-jre-headless # version 20.0.2+9-1
sudo apt install openjdk-22-jre-headless # version 22~220a-1
rzwalie@rzwalieubuntu:~$ _
```

To direct input to this VM, click inside or press Ctrl+G.

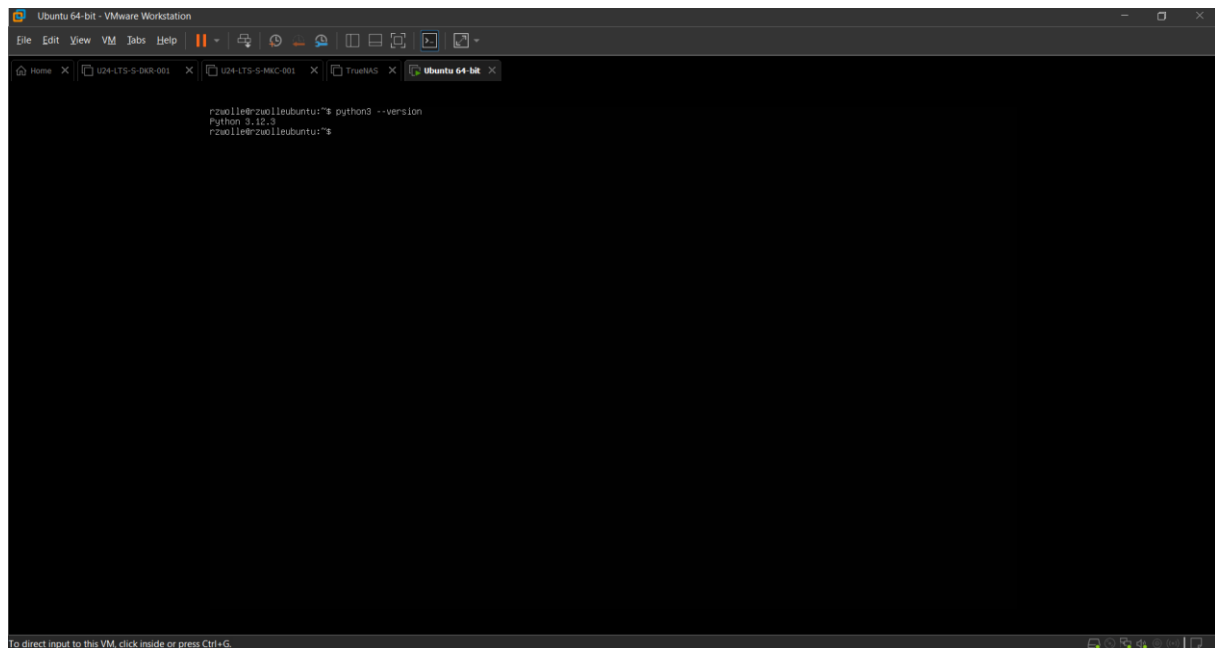
gcc --version

```
Ubuntu 64-bit - VMware Workstation
File Edit View VM Tabs Help
Home X U24-LTS-S-DKR-001 X U24-LTS-S-MMC-001 X TrueNAS X Ubuntu 64-bit X

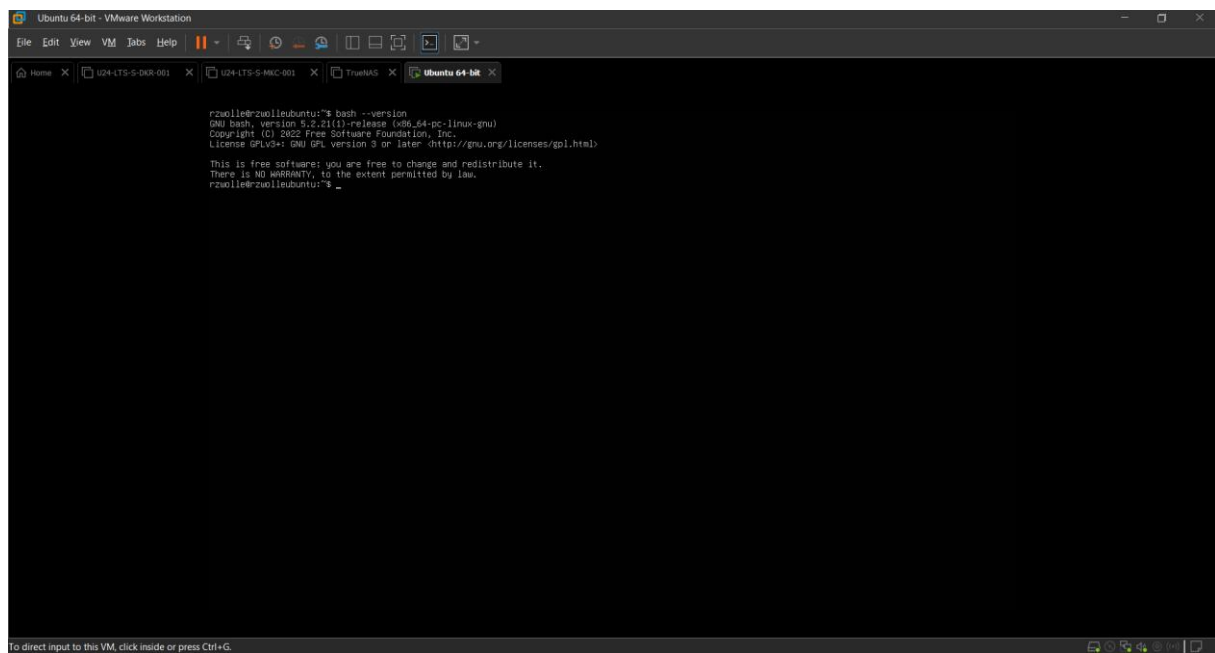
rzwalie@rzwalieubuntu:~$ gcc --version
Command 'gcc' not found, but can be installed with:
sudo apt install gcc
rzwalie@rzwalieubuntu:~$ _
```

To direct input to this VM, click inside or press Ctrl+G.

python3 --version



bash --version



Assignment 4.3: Compile

Which of the above files need to be compiled before you can run them?

Fib.c en Fibonacci.java moeten worden gecompileerd

Which source code files are compiled into machine code and then directly executable by a processor?

.C files

Which source code files are compiled to byte code?

.Java

Which source code files are interpreted by an interpreter?

.py en .sh

These source code files will perform the same calculation after compilation/interpretation. Which one is expected to do the calculation the fastest?

C.

How do I run a Java program?

1. Compilen met javac in ubuntu of met een andere compiler.
2. runnen

How do I run a Python program?

Kan gewoon gerunt worden zonder compilatie met python3 fib.py of ./fib.py

How do I run a C program?

Compileren met gcc fib.c -o fib en daarna runnen met ./fib

How do I run a Bash script?

Runnen met ./fib.sh

If I compile the above source code, will a new file be created? If so, which file?

Bij een C programma krijg je een machinecodebestand dat je kan uitvoeren en bij Java krijg je een bytecode .class file

Take relevant screenshots of the following commands:

- Compile the source files where necessary
- Make them executable
- Run them
- Which (compiled) source code file performs the calculation the fastest?

C

[illegible]

```

Adding debian:XRamp_Global_CA_Root.pem
done.
Setting up openjdk-17-jdk-headless:amd64 (17.0.17+8-1b4.04) ...
update-alternatives: using /usr/lib/jvm/java-17-openjdk-amd64/bin/jar to provide /usr/bin/jar (jar) in auto mode
update-alternatives: using /usr/lib/jvm/java-17-openjdk-amd64/bin/jarsigner to provide /usr/bin/jarsigner (jarsigner) in auto mode
update-alternatives: using /usr/lib/jvm/java-17-openjdk-amd64/bin/javadoc to provide /usr/bin/javadoc (javadoc) in auto mode
update-alternatives: using /usr/lib/jvm/java-17-openjdk-amd64/bin/javap to provide /usr/bin/javap (javap) in auto mode
update-alternatives: using /usr/lib/jvm/java-17-openjdk-amd64/bin/jcsh to provide /usr/bin/jcsh (jcsh) in auto mode
update-alternatives: using /usr/lib/jvm/java-17-openjdk-amd64/bin/jdb to provide /usr/bin/jdb (jdb) in auto mode
update-alternatives: using /usr/lib/jvm/java-17-openjdk-amd64/bin/jdeps to provide /usr/bin/jdeps (jdeps) in auto mode
update-alternatives: using /usr/lib/jvm/java-17-openjdk-amd64/bin/jfr to provide /usr/bin/jfr (jfr) in auto mode
update-alternatives: using /usr/lib/jvm/java-17-openjdk-amd64/bin/jimage to provide /usr/bin/jimage (jimage) in auto mode
update-alternatives: using /usr/lib/jvm/java-17-openjdk-amd64/bin/jinfo to provide /usr/bin/jinfo (jinfo) in auto mode
update-alternatives: using /usr/lib/jvm/java-17-openjdk-amd64/bin/jlink to provide /usr/bin/jlink (jlink) in auto mode
update-alternatives: using /usr/lib/jvm/java-17-openjdk-amd64/bin/jmap to provide /usr/bin/jmap (jmap) in auto mode
update-alternatives: using /usr/lib/jvm/java-17-openjdk-amd64/bin/jmod to provide /usr/bin/jmod (jmod) in auto mode
update-alternatives: using /usr/lib/jvm/java-17-openjdk-amd64/bin/jps to provide /usr/bin/jps (jps) in auto mode
update-alternatives: using /usr/lib/jvm/java-17-openjdk-amd64/bin/jrunscript to provide /usr/bin/jrunscript (jrunscript) in auto mode
update-alternatives: using /usr/lib/jvm/java-17-openjdk-amd64/bin/jshell to provide /usr/bin/jshell (jshell) in auto mode
update-alternatives: using /usr/lib/jvm/java-17-openjdk-amd64/bin/jstack to provide /usr/bin/jstack (jstack) in auto mode
update-alternatives: using /usr/lib/jvm/java-17-openjdk-amd64/bin/jstat to provide /usr/bin/jstat (jstat) in auto mode
update-alternatives: using /usr/lib/jvm/java-17-openjdk-amd64/bin/jstatd to provide /usr/bin/jstatd (jstatd) in auto mode
update-alternatives: using /usr/lib/jvm/java-17-openjdk-amd64/bin/serialver to provide /usr/bin/serialver (serialver) in auto mode
update-alternatives: using /usr/lib/jvm/java-17-openjdk-amd64/bin/jhsdb to provide /usr/bin/jhsdb (jhsdb) in auto mode
Scanning processes...
Scanning linux images...

Running kernel seems to be up-to-date.

No services need to be restarted.

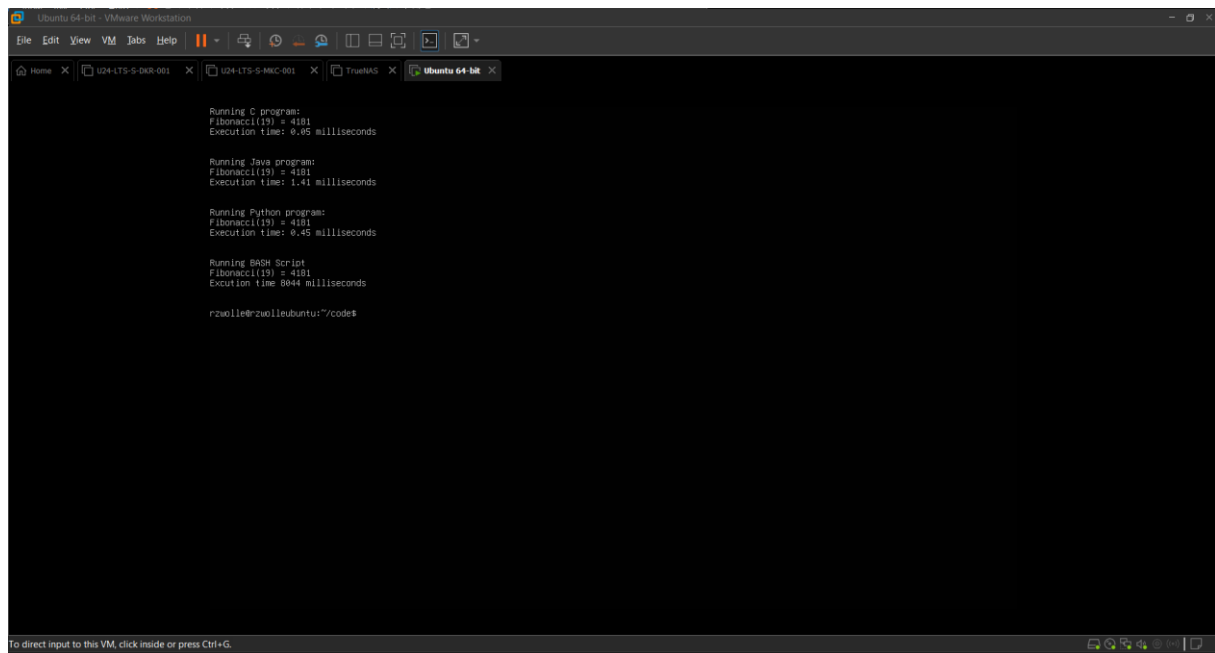
No containers need to be restarted.

No user sessions are running outdated binaries.

No VM guests are running outdated hypervisor (qemu) binaries on this host.
rzwlle@zwlleubuntu:/code$ ls
fib.c  fib.h  fib.o  fib.py  fib.sh  rzwlle.sh
rzwlle@zwlleubuntu:/code$ javac Fibonacci.java
rzwlle@zwlleubuntu:/code$ ls
fib.c  Fibonacci.class  Fibonacci.java  fib.py  fib.sh  rzwlle.sh
rzwlle@zwlleubuntu:/code$ java Fibonacci.class
Error: Could not find or load main class Fibonacci.class
Caused by: java.lang.ClassNotFoundException: Fibonacci.class
rzwlle@zwlleubuntu:/code$ java Fibonacci
Fibonacci(8) = 256
Execution time: 0.64 milliseconds
rzwlle@zwlleubuntu:/code$

```

Runall.sh (compiled en runt alle bestanden via bash script)



The screenshot shows a VMware Workstation window titled 'Ubuntu 64-bit - VMware Workstation'. The interface includes a menu bar (File, Edit, View, VM, Tabs, Help) and a toolbar. Below the toolbar is a tab bar with four tabs: 'Home', 'U24-LTS-S-DNR-001', 'U24-LTS-S-MMC-001', and 'TrueNAS'. The 'U24-LTS-S-MMC-001' tab is active, displaying a terminal window. The terminal output shows the execution of a script named 'Runall.sh' which runs four programs: C, Java, Python, and a BASH script. Each program calculates the 19th Fibonacci number (4181) and reports its execution time. The BASH script's execution time is significantly longer (8044 milliseconds) than the others. The terminal prompt at the bottom is 'rzwalle@zuolleubuntu:~/codes\$'. At the very bottom of the VMware window, a status bar reads 'To direct input to this VM, click inside or press Ctrl+G.' and contains several icons.

```
Running C program:
Fibonacci(19) = 4181
Execution time: 0.05 milliseconds

Running Java program:
Fibonacci(19) = 4181
Execution time: 1.41 milliseconds

Running Python program:
Fibonacci(19) = 4181
Execution time: 0.45 milliseconds

Running BASH Script
Fibonacci(19) = 4181
Execution time 8044 milliseconds

rzwalle@zuolleubuntu:~/codes$
```

Assignment 4.4: Optimize

Take relevant screenshots of the following commands:

- a) Figure out which parameters you need to pass to **the gcc** compiler so that the compiler performs a number of optimizations that will ensure that the compiled source code will run faster. **Tip!** The parameters are usually a letter followed by a number. Also read **page 191** of your book, but find a better optimization in the man pages. Please note that Linux is case sensitive.
- b) Compile **fib.c** again with the optimization parameters
- c) Run the newly compiled program. Is it true that it now performs the calculation faster?
- d) Edit the file **runall.sh**, so you can perform all four calculations in a row using this Bash script. So the (compiled/interpreted) C, Java, Python and Bash versions of Fibonacci one after the other.

Assignment 4.5: More ARM Assembly

Like the factorial example, you can also implement the calculation of a power of 2 in assembly. For example you want to calculate $2^4 = 16$. Use iteration to calculate the result. Store the result in r0.

Main:

```
mov r1, #2
```

```
mov r2, #4
```

Loop:

End:

Complete the code. See the PowerPoint slides of week 4.

Screenshot of the completed code here.

```
1 Main:
2     MOV r1, #2
3     MOV r2, #4
4     MOV r0, #1
5
6 Loop:
7     MUL r0, r0, r1
8     SUB r2, r2, #1
9     CMP r2, #0
10    BEQ End
11    B Loop
12
13 End:
```

Ready? Save this file and export it as a pdf file with the name: [week4.pdf](#)