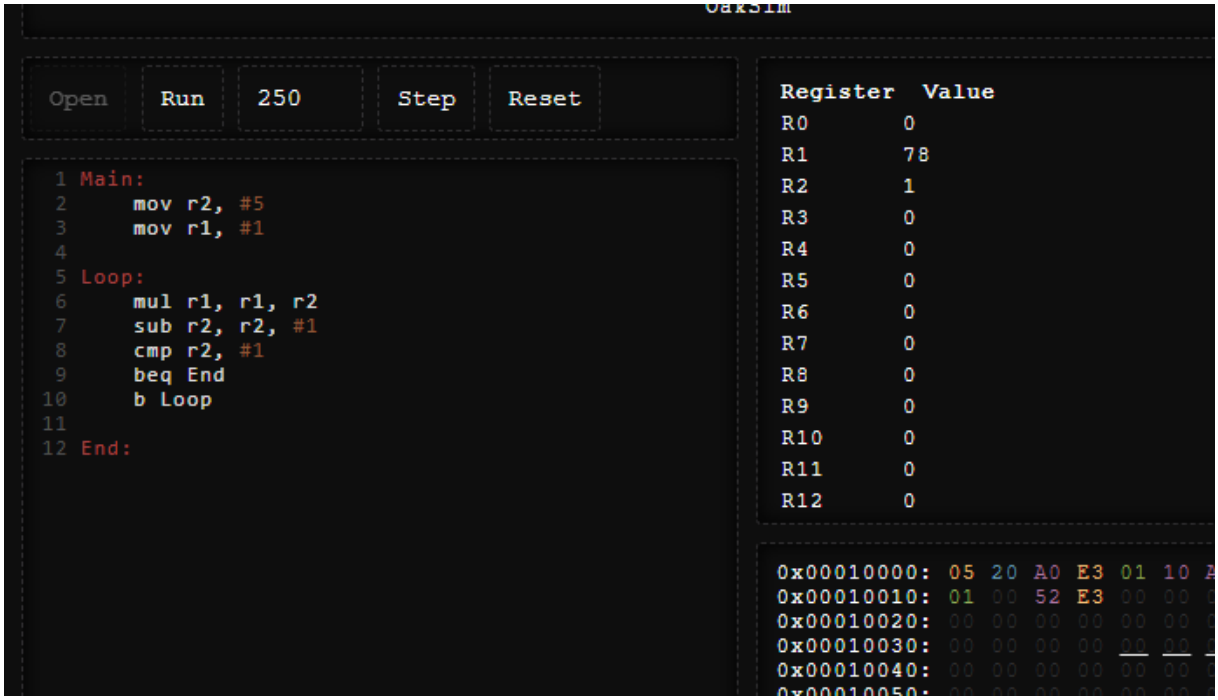
Template Week 4 – Software

Student number: 569073

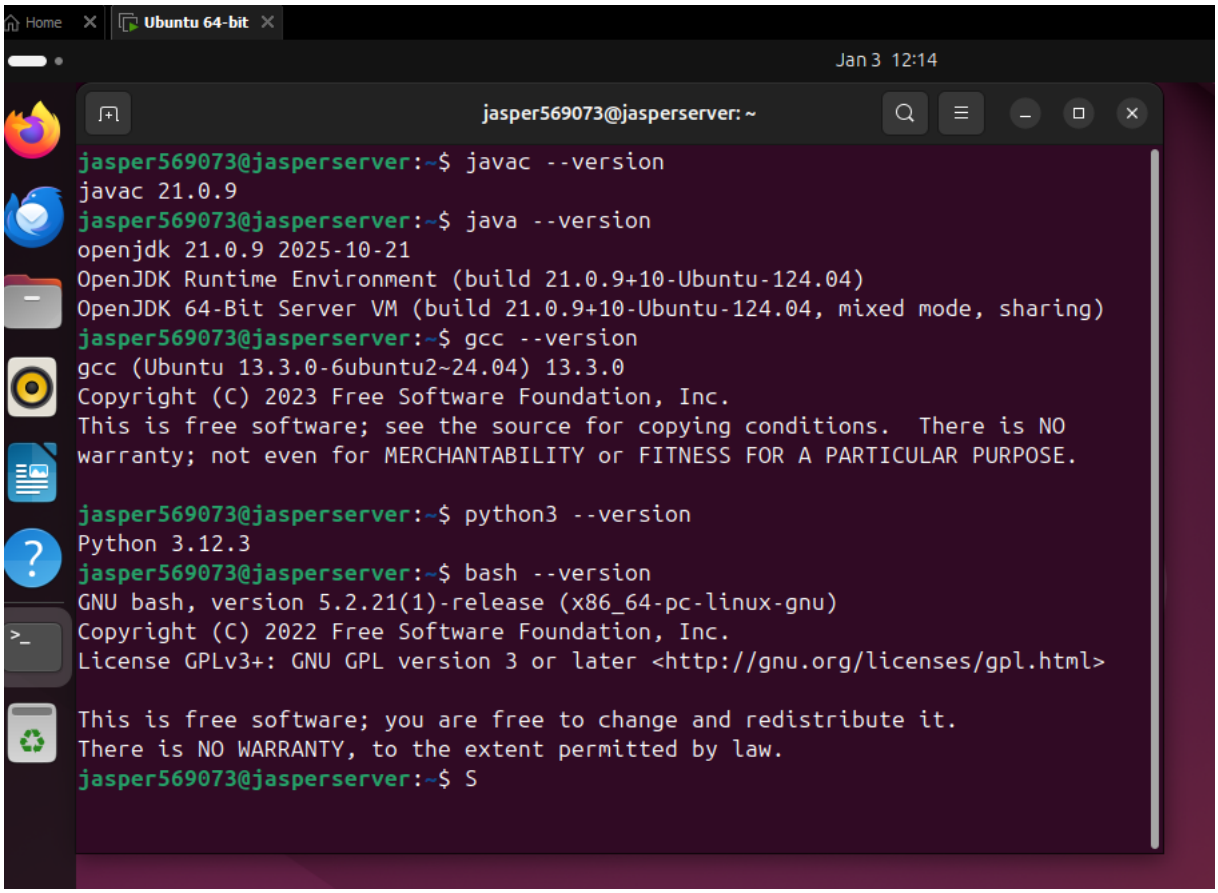
Assignment 4.1: ARM assembly

Screenshot of working assembly code of factorial calculation:



Assignment 4.2: Programming languages

Take screenshots that the following commands work:



javac --version

java --version

gcc --version

python3 --version

bash --version

Assignment 4.3: Compile

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

Fibonacci.java and fib.c are compiled files the other two are interpreted

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

Only fib.c is compiled into machine code.

Which source code files are compiled to byte code?

Fibonacci.java is compiled to byte code

Which source code files are interpreted by an interpreter?

fib.py and fib.sh are interpreted.

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

fib.c because it's a compiled language

How do I run a Java program?

Java programs run on a java virtual machine

On pc Java programs can be compiled and run using a JDK

For example "java program.java"

How do I run a Python program?

Python programs are run using the Python interpreter

For example "python program.py"

How do I run a C program?

A C program is compiled using a C compiler and then the compiled program can be executed

For example "gcc program.c -o program" then in terminal "./program"

gcc is one of the most common c compilers

How do I run a Bash script?

First u need to make the script executable then It can be ran using bash

For example to make executable “chmod +x program.sh” and then “ ./program.sh”

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

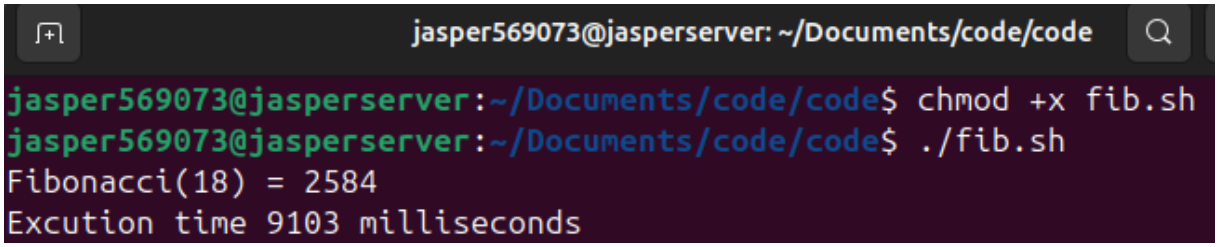
For java a new file gets created it will be a .jar file

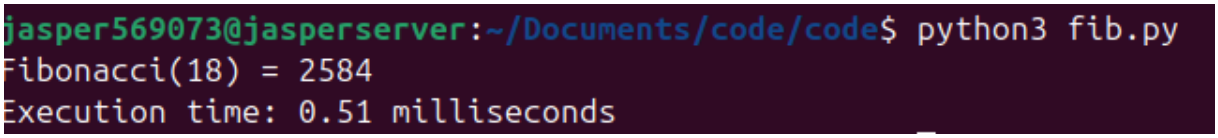
For c a new file gets created it will be a .exe file

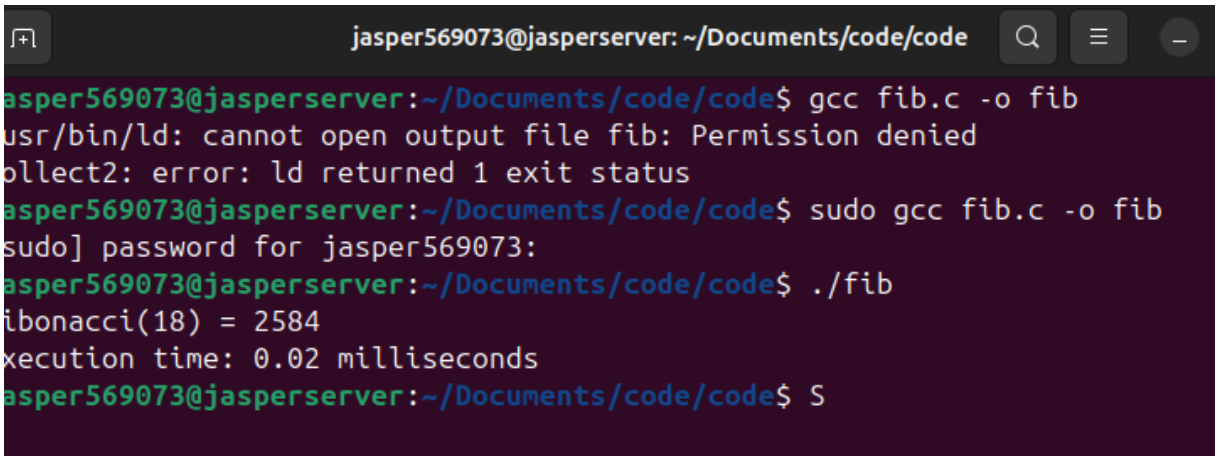
Take relevant screenshots of the following commands:

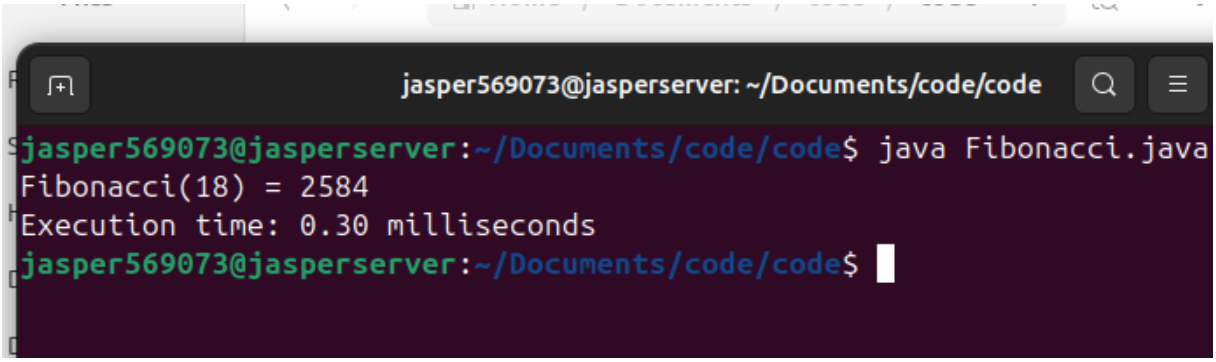
- Compile the source files where necessary
- Make them executable
- Run them
- Which (compiled) source code file performs the calculation the fastest?

C is the fastest







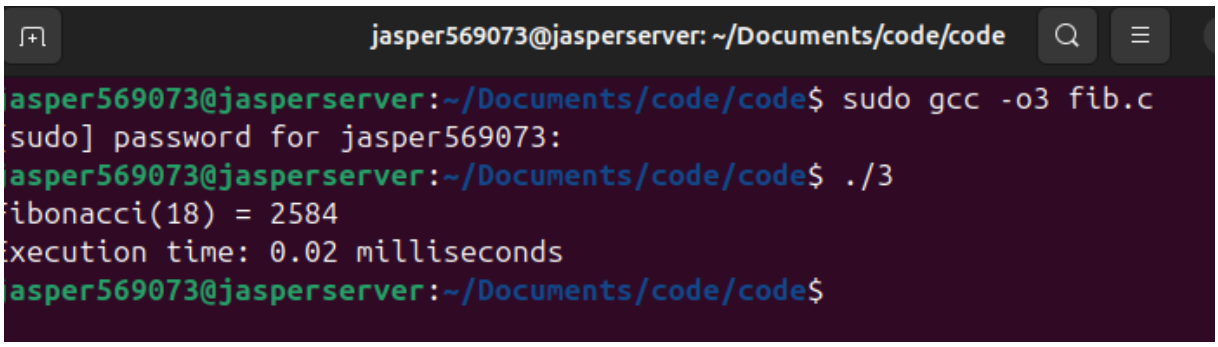


Assignment 4.4: Optimize

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.
-o3

b) Compile **fib.c** again with the optimization parameters

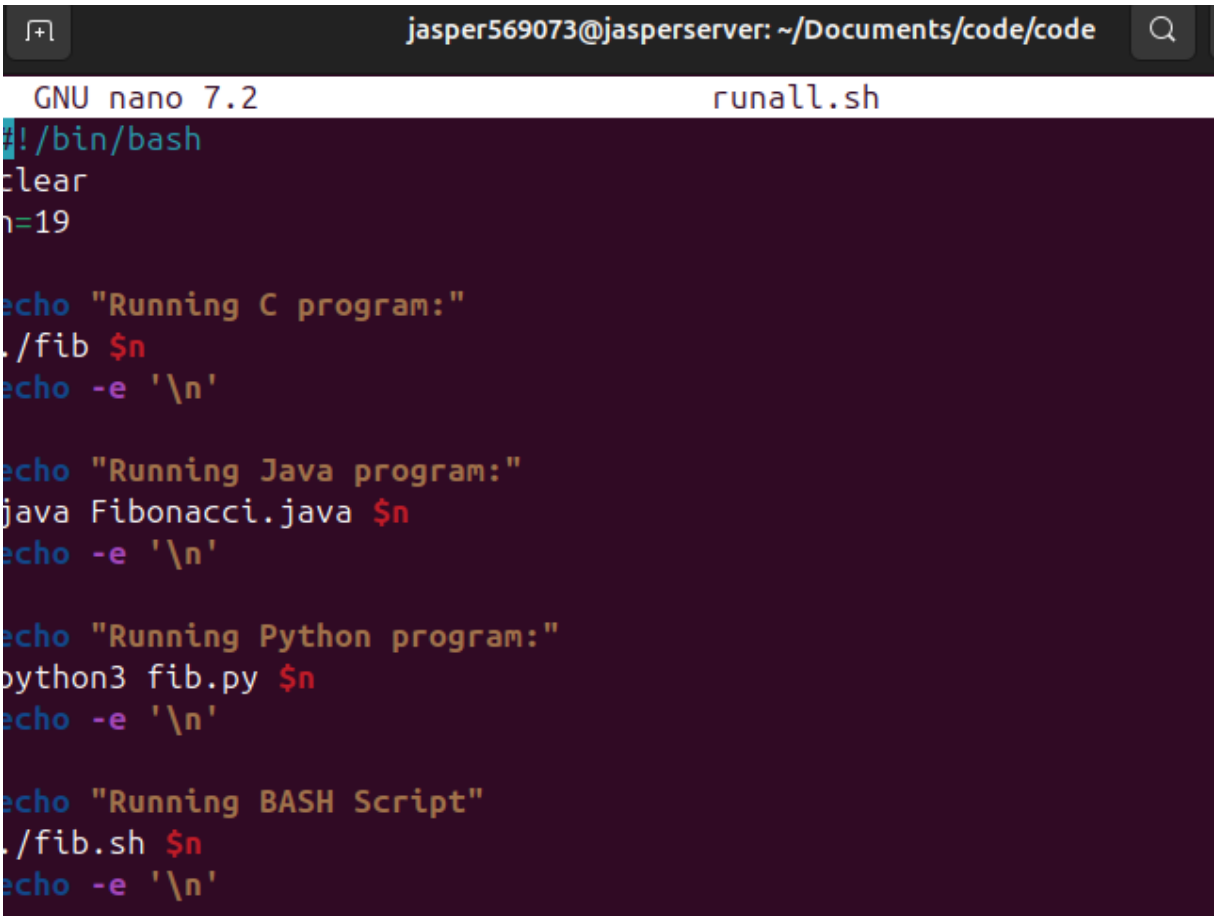


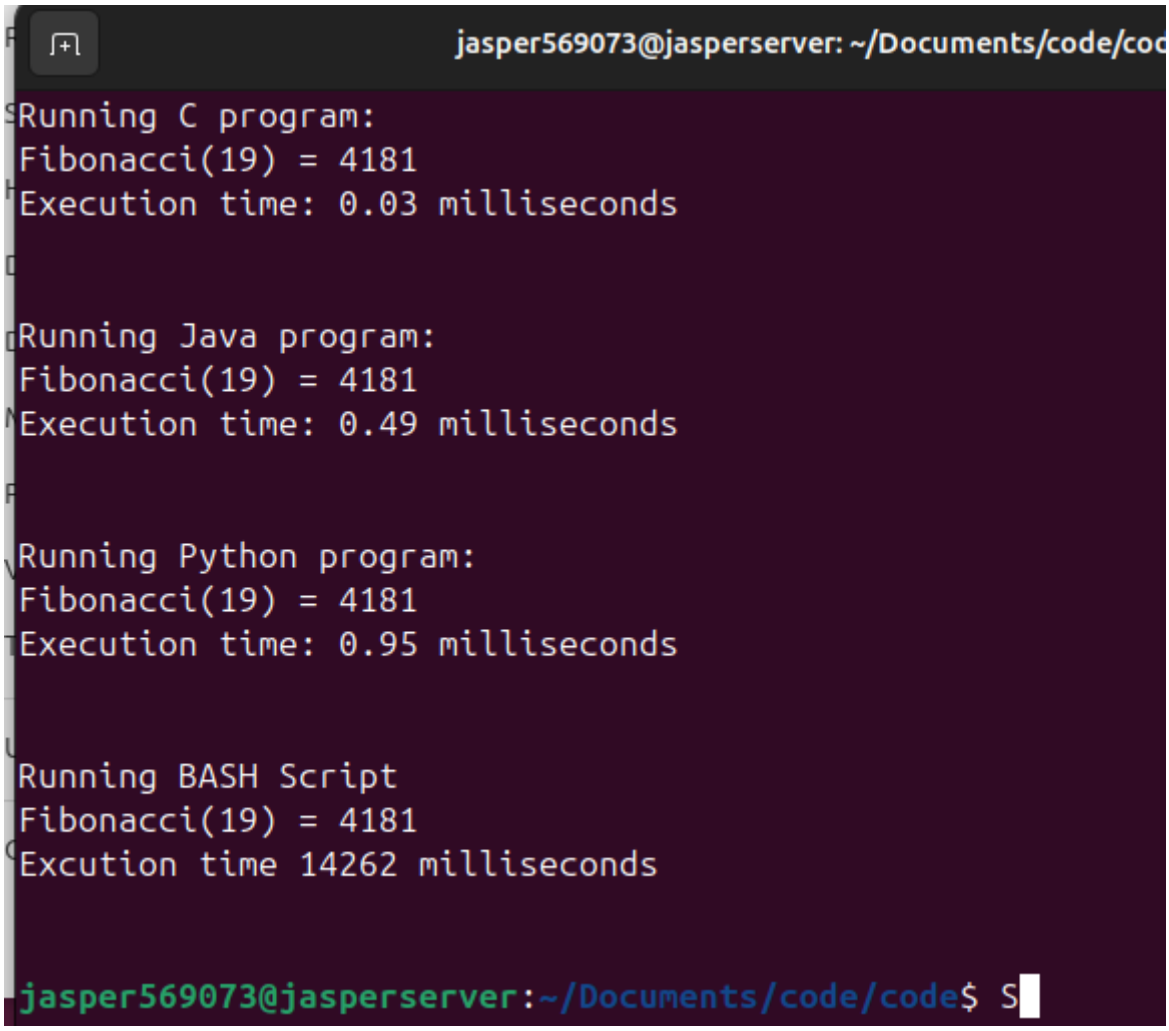
c) Run the newly compiled program. Is it true that it now performs the calculation faster?

It's the exact same time so it doesn't seem faster with this program

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.

e)


jasper569073@jasperserver: ~/Documents/code/code
GNU nano 7.2 runall.sh
#!/bin/bash
clear
n=19
echo "Running C program:"
./fib $n
echo -e '\n'
echo "Running Java program:"
java Fibonacci.java $n
echo -e '\n'
echo "Running Python program:"
python3 fib.py $n
echo -e '\n'
echo "Running BASH Script"
./fib.sh $n
echo -e '\n'


jasper569073@jasperserver: ~/Documents/code/coc
Running C program:
Fibonacci(19) = 4181
Execution time: 0.03 milliseconds
Running Java program:
Fibonacci(19) = 4181
Execution time: 0.49 milliseconds
Running Python program:
Fibonacci(19) = 4181
Execution time: 0.95 milliseconds
Running BASH Script
Fibonacci(19) = 4181
Excution time 14262 milliseconds
jasper569073@jasperserver:~/Documents/code/code$ S

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.

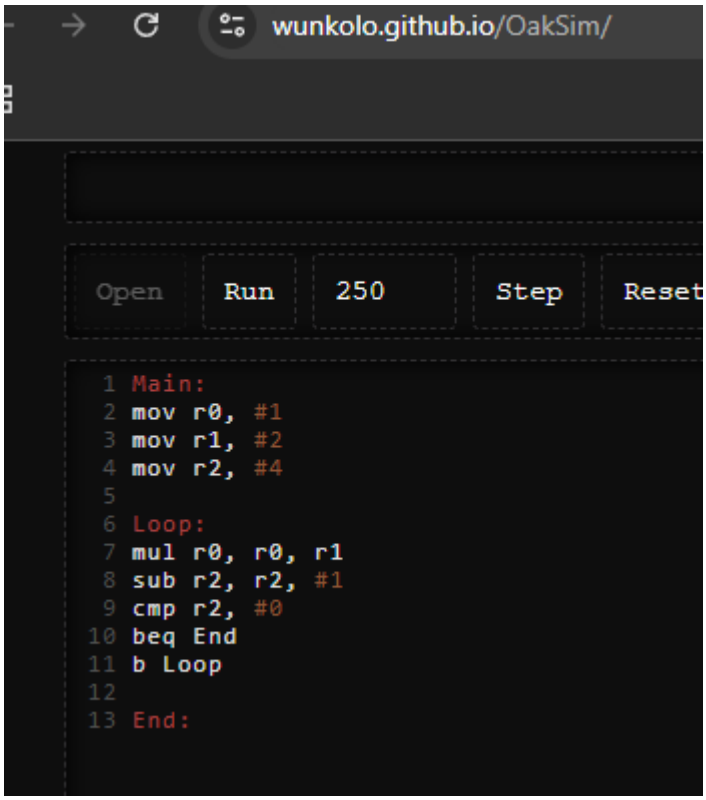
```
Main:
mov r1, #2
mov r2, #4

Loop:

End:
```

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

Screenshot of the completed code here.



Ready? Save this file and export it as a pdf file with the name: **week4.pdf**