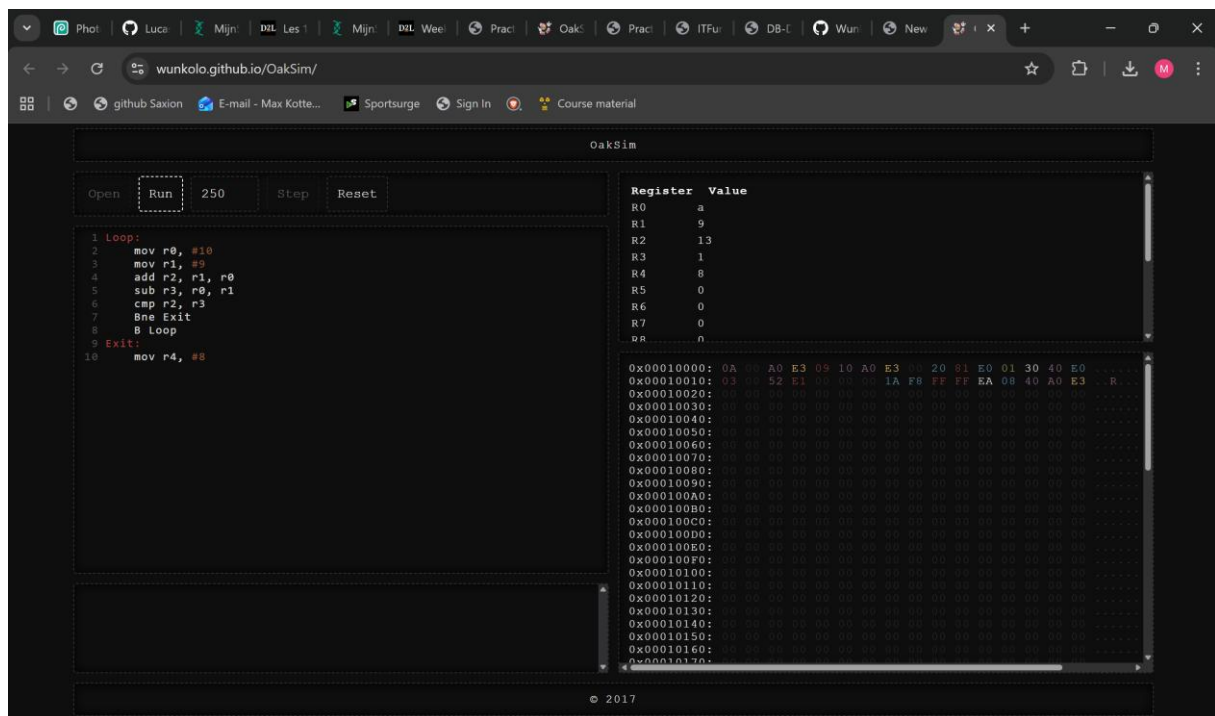


# Template Week 4 – Software

Student number:

## Assignment 4.1: ARM assembly

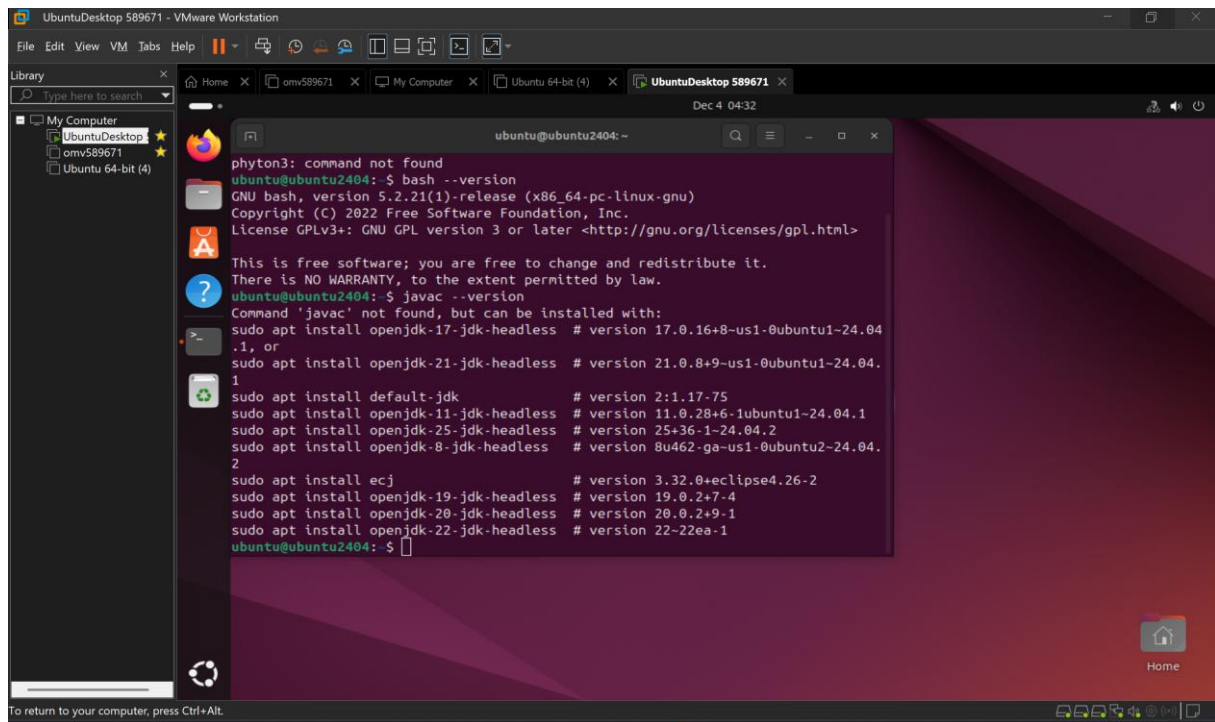
Screenshot of working assembly code of factorial calculation:



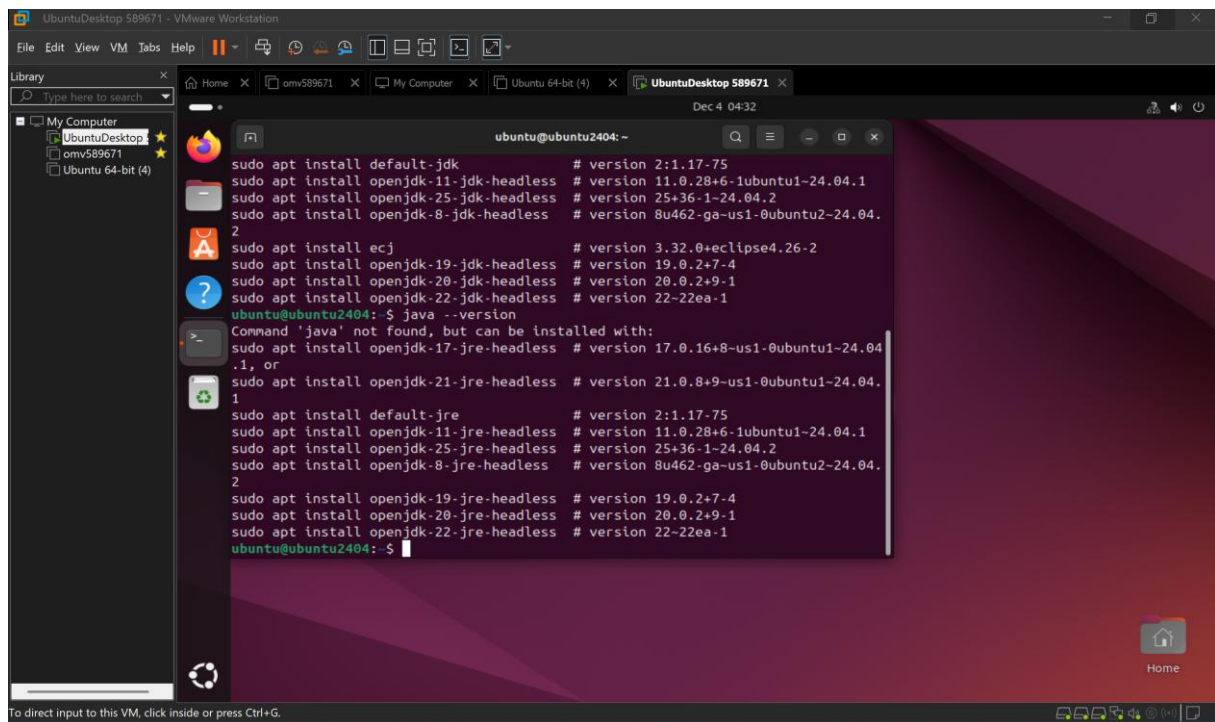
## Assignment 4.2: Programming languages

Take screenshots that the following commands work:

javac -version



java --version



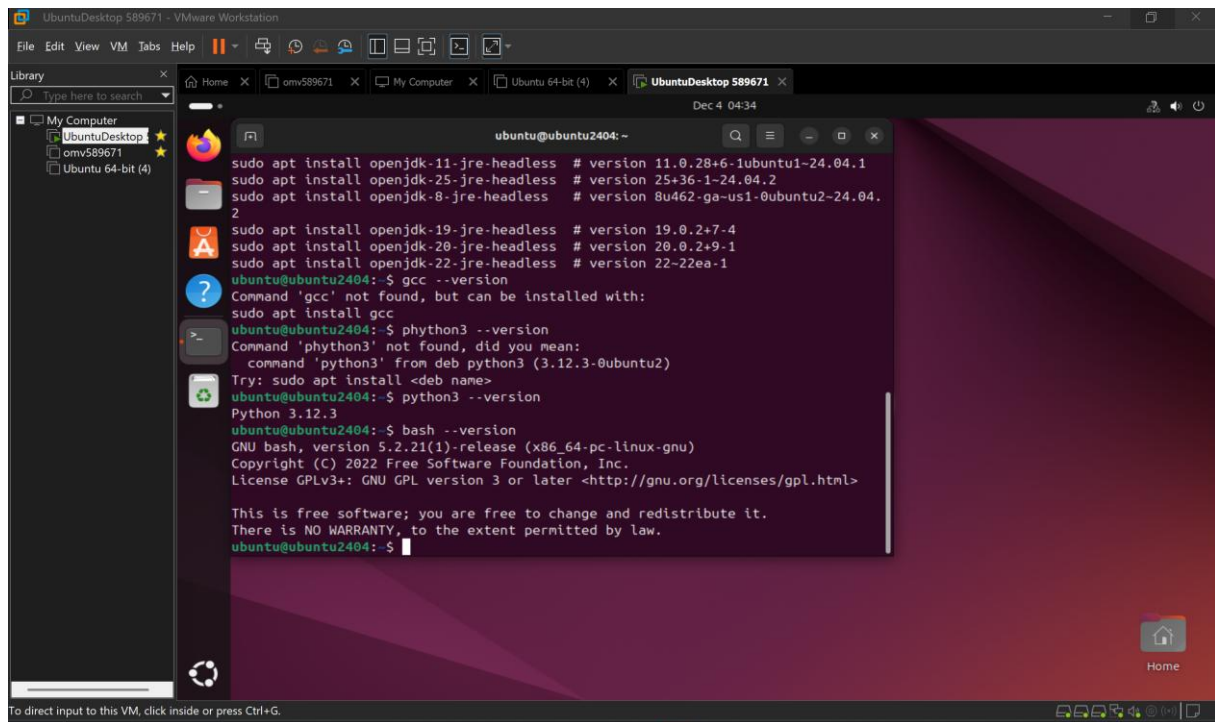
gcc --version

```
ubuntu@ubuntu2404:~$ sudo apt install openjdk-8-jdk-headless # version 8u462-ga-us1-0ubuntu2-24.04.2
ubuntu@ubuntu2404:~$ sudo apt install ecj # version 3.32.0+eclipse4.26-2
ubuntu@ubuntu2404:~$ sudo apt install openjdk-19-jdk-headless # version 19.0.2+7-4
ubuntu@ubuntu2404:~$ sudo apt install openjdk-20-jdk-headless # version 20.0.2+9-1
ubuntu@ubuntu2404:~$ sudo apt install openjdk-22-jdk-headless # version 22-22ea-1
ubuntu@ubuntu2404:~$ java --version
Command 'java' not found, but can be installed with:
ubuntu@ubuntu2404:~$ sudo apt install openjdk-17-jre-headless # version 17.0.16+8-us1-0ubuntu1-24.04.1, or
ubuntu@ubuntu2404:~$ sudo apt install openjdk-21-jre-headless # version 21.0.8+9-us1-0ubuntu1-24.04.1
ubuntu@ubuntu2404:~$ sudo apt install default-jre # version 2:1.17-75
ubuntu@ubuntu2404:~$ sudo apt install openjdk-11-jre-headless # version 11.0.28+6-1ubuntu1-24.04.1
ubuntu@ubuntu2404:~$ sudo apt install openjdk-25-jre-headless # version 25+36-1-24.04.2
ubuntu@ubuntu2404:~$ sudo apt install openjdk-8-jre-headless # version 8u462-ga-us1-0ubuntu2-24.04.2
ubuntu@ubuntu2404:~$ sudo apt install openjdk-19-jre-headless # version 19.0.2+7-4
ubuntu@ubuntu2404:~$ sudo apt install openjdk-20-jre-headless # version 20.0.2+9-1
ubuntu@ubuntu2404:~$ sudo apt install openjdk-22-jre-headless # version 22-22ea-1
ubuntu@ubuntu2404:~$ gcc --version
Command 'gcc' not found, but can be installed with:
ubuntu@ubuntu2404:~$ sudo apt install gcc
ubuntu@ubuntu2404:~$
```

python3 --version

```
ubuntu@ubuntu2404:~$ sudo apt install python3
ubuntu@ubuntu2404:~$ python3 --version
Python 3.12.3
ubuntu@ubuntu2404:~$
```

bash --version



### Assignment 4.3: Compile

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

**Javac en GCC. De andere zijn interpreters.**

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

**Gcc. Alle andere gebruiken iets van een tussenstap voordat het gerunt kan worden.**

Which source code files are compiled to byte code?

**Javac. Dit compilet het zodat java het kan runnen**

Which source code files are interpreted by an interpreter?

Python en bash

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

**Gcc, deze praat direct met de hardware zonder tussen stap**

How do I run a Java program?

**Een .java bestand word gecompileerd door een java compiler. Deze gegevens worden omgezet naar bytecode die de can gerunt word op een JVM.**

How do I run a Python program?

**Een .py bestand staat ergens op de computer. Dan can deze gerunt worden door hem met python op te starten. Bijv python3 helloworld.py**

How do I run a C program?

.c bestand word gerunt. Door GCC. Gcc maakt een uitvoerbaar bestand

How do I run a Bash script?

Een Bash-script maak je door een bestand met de extensie .sh te schrijven en bovenaan #!/bin/bash te zetten. Daarna geef je het uitvoerrechten met chmod +x script.sh en start je het met ./script.sh.

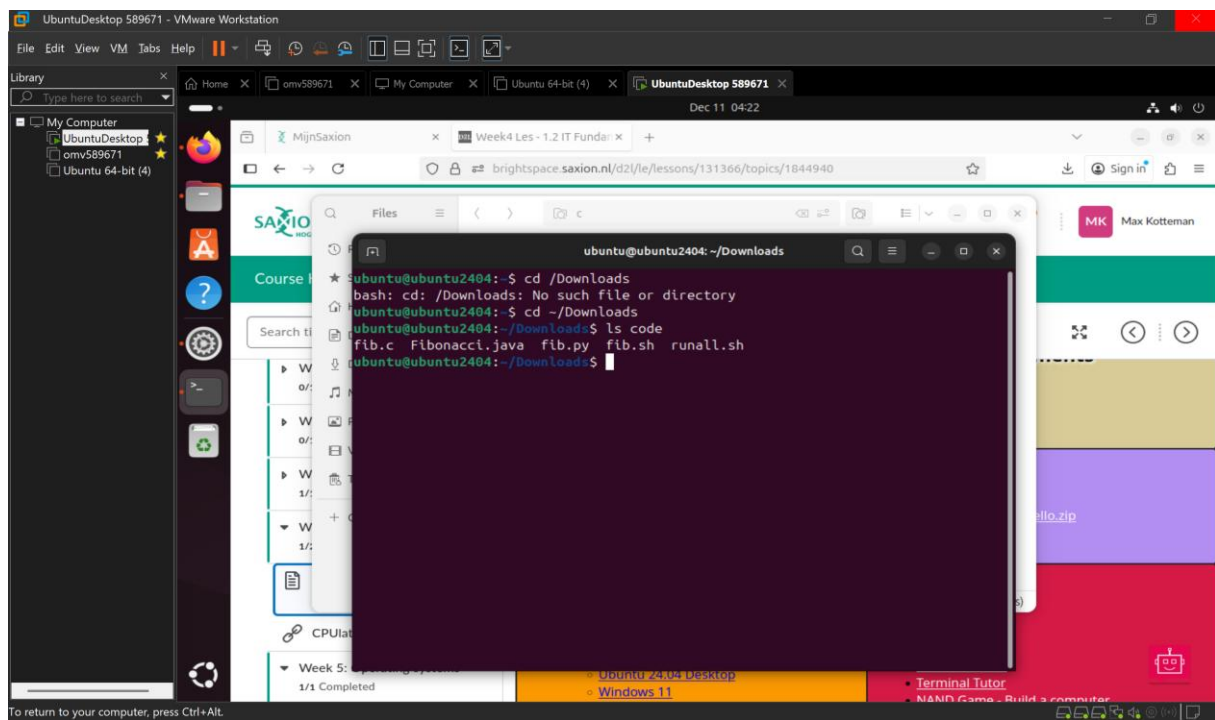
If I compile the above source code, will a new file be created? If so, which file?

Ja, nieuwe bestanden ontstaan bij Java , C en soms bij Python.

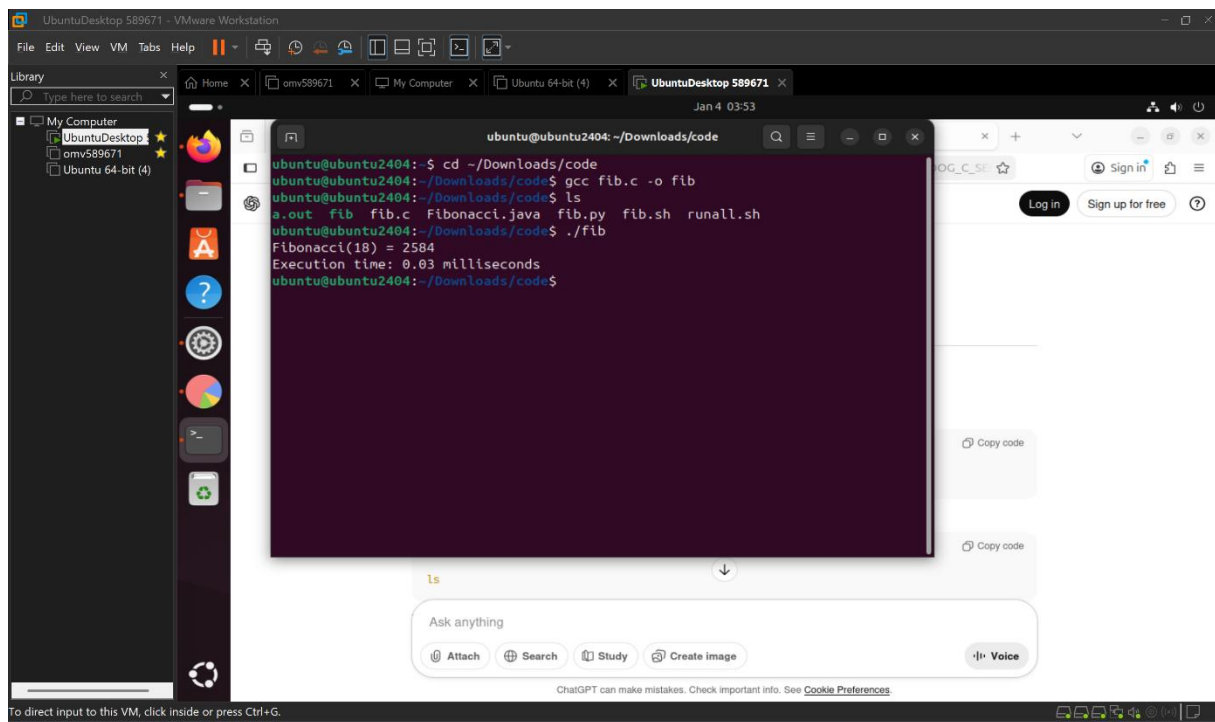
Take relevant screenshots of the following commands:

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

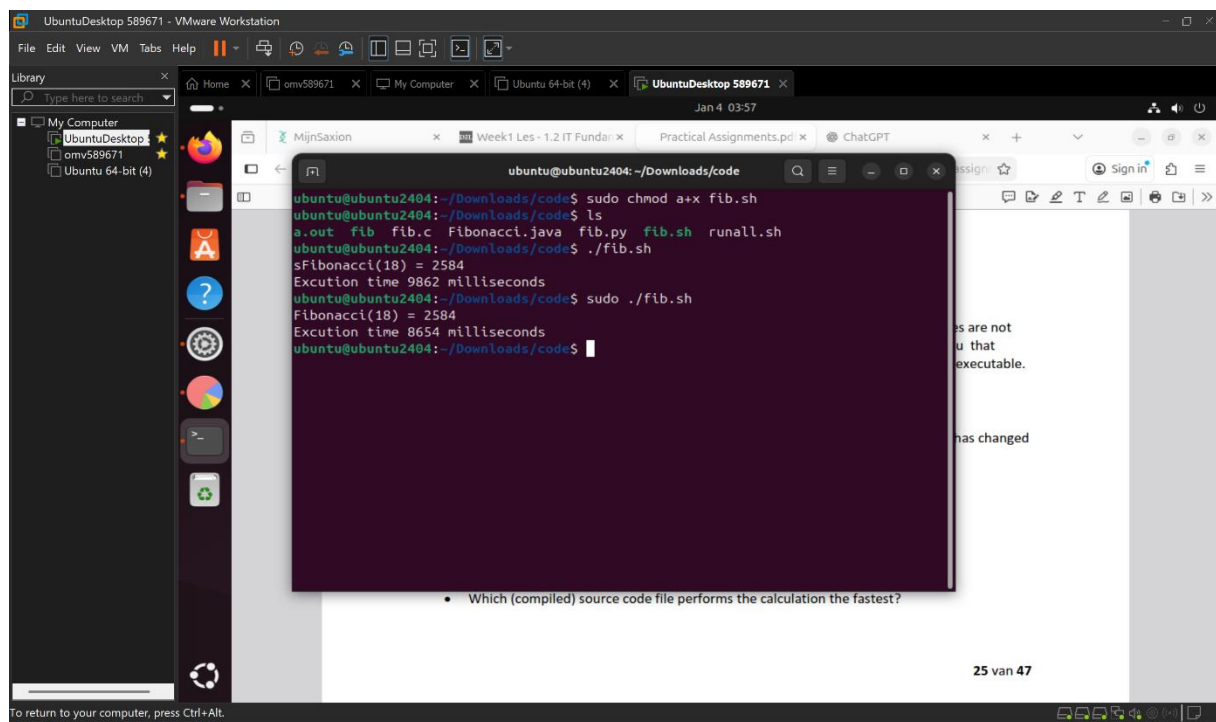




C



## BASH

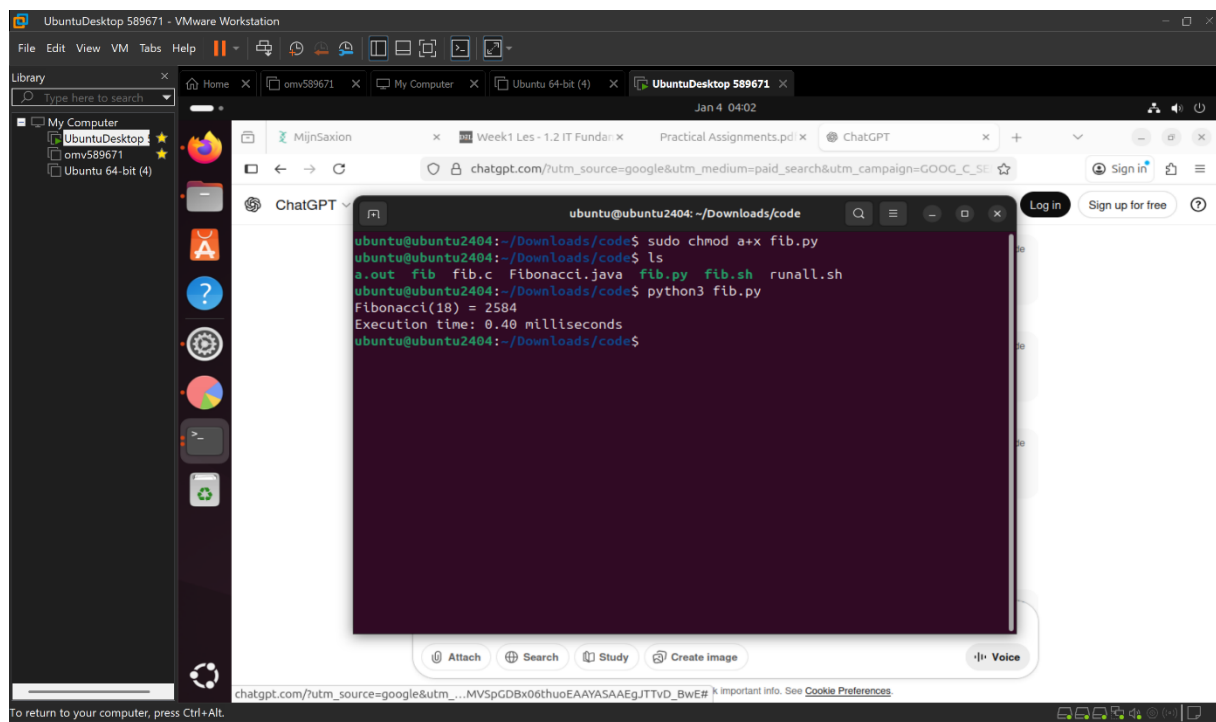


The screenshot shows a VMware Workstation window titled 'UbuntuDesktop 589671 - VMware Workstation'. The main window displays a terminal window titled 'ubuntu@ubuntu2404: ~/Downloads/code'. The terminal shows the following commands and output:

```
ubuntu@ubuntu2404:~/Downloads/code$ sudo chmod a+x fib.sh
ubuntu@ubuntu2404:~/Downloads/code$ ls
a.out fib fib.c Fibonacci.java fib.py fib.sh runall.sh
ubuntu@ubuntu2404:~/Downloads/code$ ./fib.sh
sFibonacci(18) = 2584
Execution time 9862 milliseconds
ubuntu@ubuntu2404:~/Downloads/code$ sudo ./fib.sh
Fibonacci(18) = 2584
Execution time 8654 milliseconds
ubuntu@ubuntu2404:~/Downloads/code$
```

The terminal window is overlaid on a web browser window showing 'chatgpt.com/rutm\_source=google&utm\_medium=paid\_search&utm\_campaign=GOOG\_C\_SE'. The browser window also shows a 'Sign in' button and a 'Log in' button. The terminal window is titled 'ubuntu@ubuntu2404: ~/Downloads/code'.

## Python

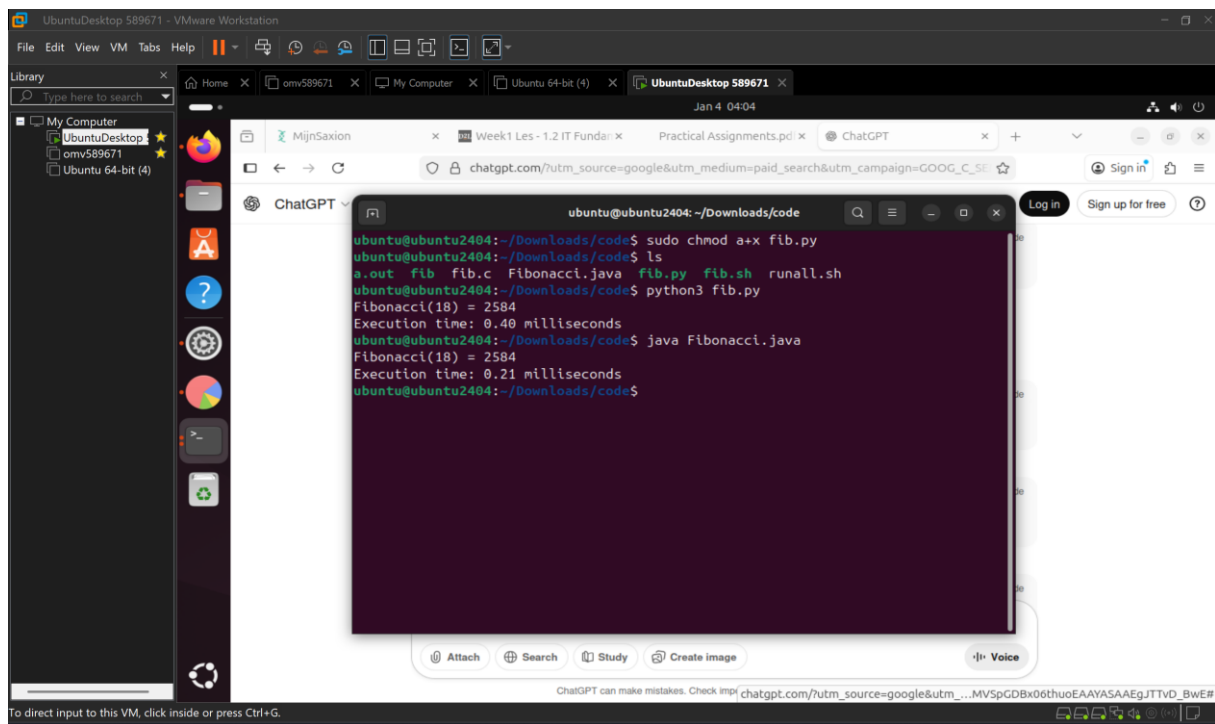


The screenshot shows a VMware Workstation window titled 'UbuntuDesktop 589671 - VMware Workstation'. The main window displays a terminal window titled 'ubuntu@ubuntu2404: ~/Downloads/code'. The terminal shows the following commands and output:

```
ubuntu@ubuntu2404:~/Downloads/code$ sudo chmod a+x fib.py
ubuntu@ubuntu2404:~/Downloads/code$ ls
a.out fib fib.c Fibonacci.java fib.py fib.sh runall.sh
ubuntu@ubuntu2404:~/Downloads/code$ python3 fib.py
Fibonacci(18) = 2584
Execution time: 0.48 milliseconds
ubuntu@ubuntu2404:~/Downloads/code$
```

The terminal window is overlaid on a web browser window showing 'chatgpt.com/rutm\_source=google&utm\_medium=paid\_search&utm\_campaign=GOOG\_C\_SE'. The browser window also shows a 'Sign in' button and a 'Log in' button. The terminal window is titled 'ubuntu@ubuntu2404: ~/Downloads/code'.

# Java





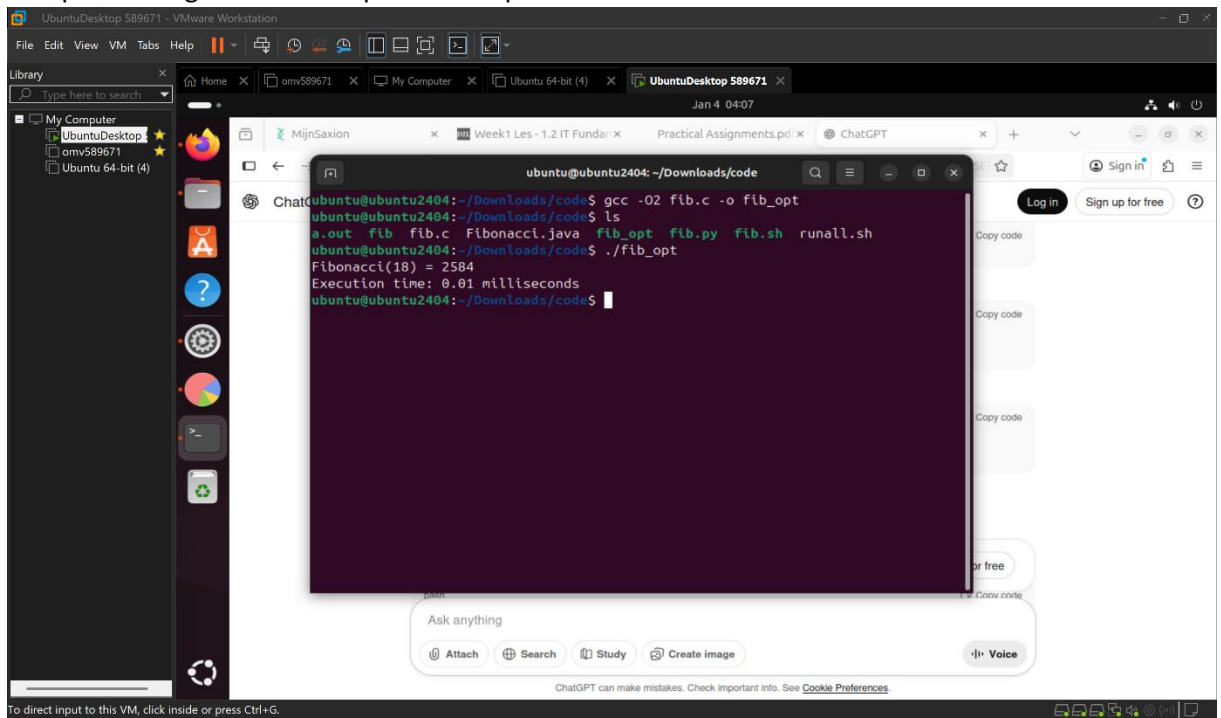
## Assignment 4.4: Optimize

Take relevant screenshots of the following commands:

- 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.

-O2

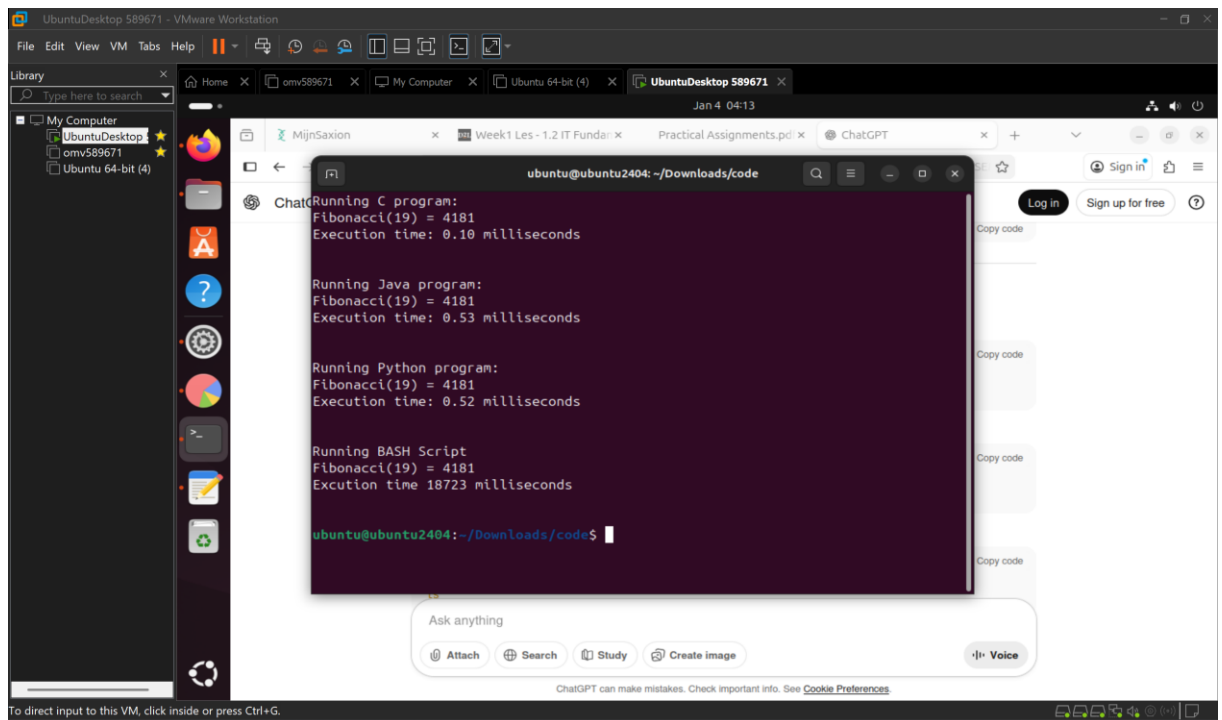
- Compile **fib.c** again with the optimization parameters



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

Ja. Hij is veel sneller

- 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
```

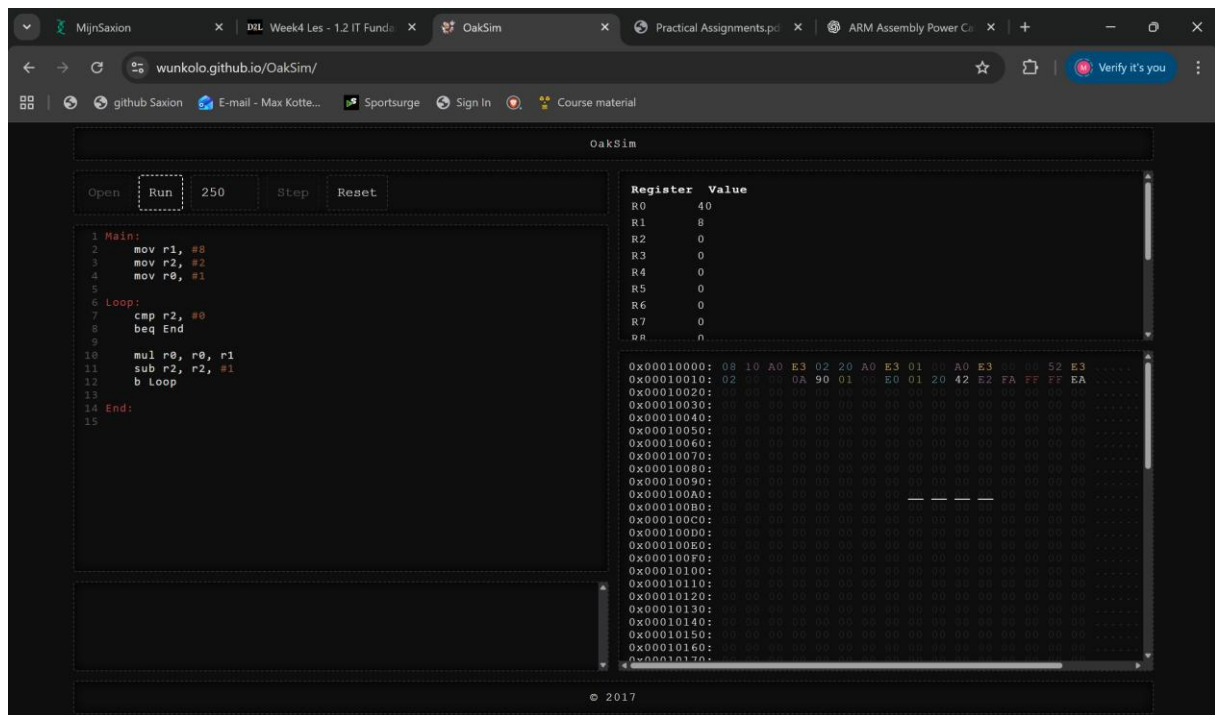
```
mov r2, #4
```

Loop:

End:

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

Screenshot of the completed code here.



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