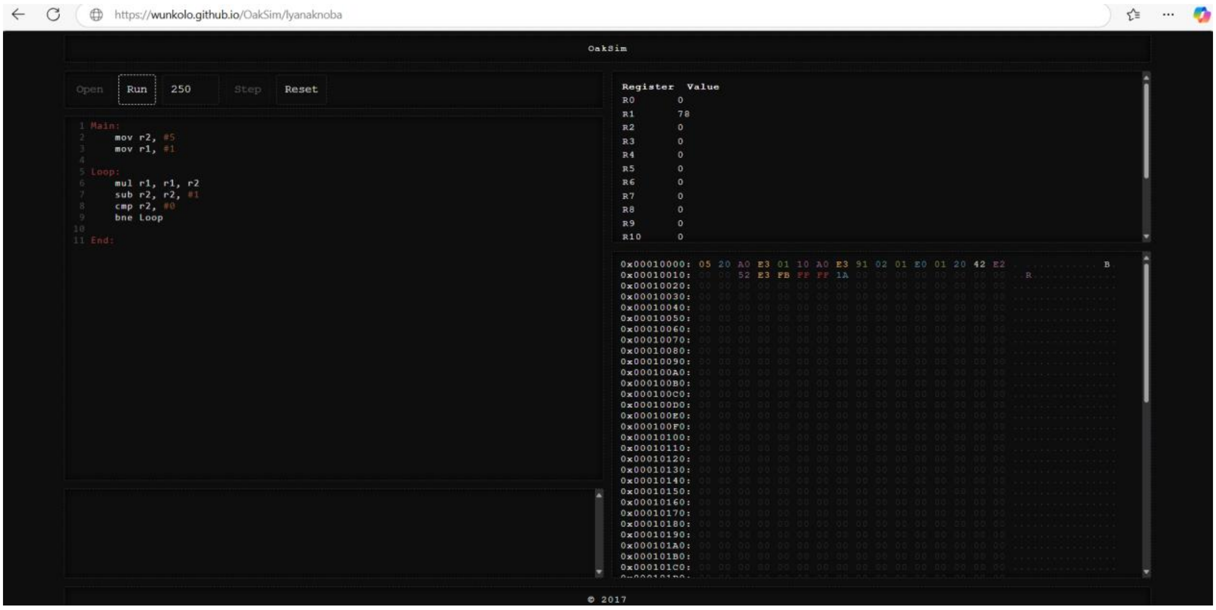
Template Week 4 – Software

Student number: 575933

Assignment 4.1: ARM assembly

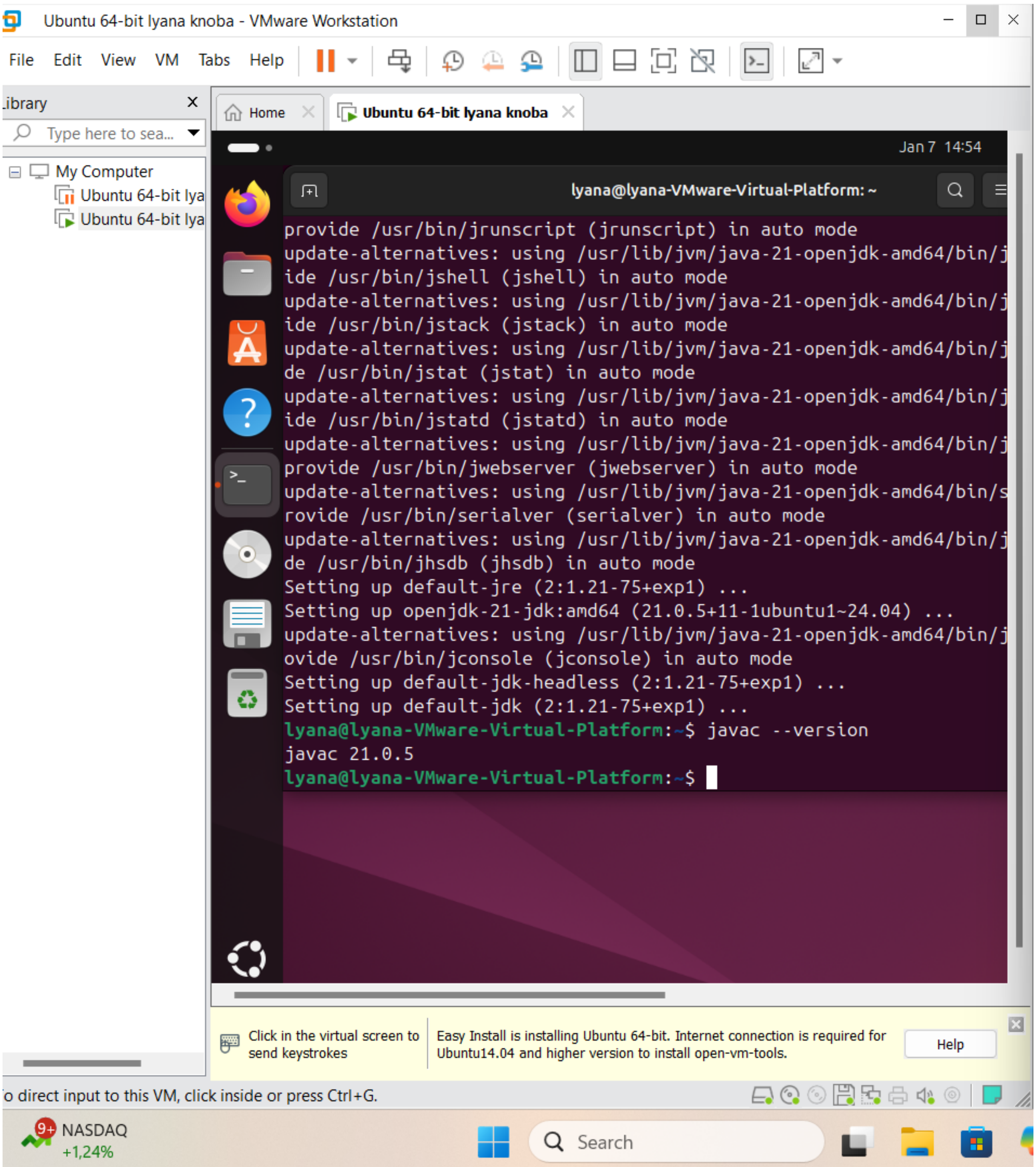
Screenshot of working assembly code of factorial calculation:



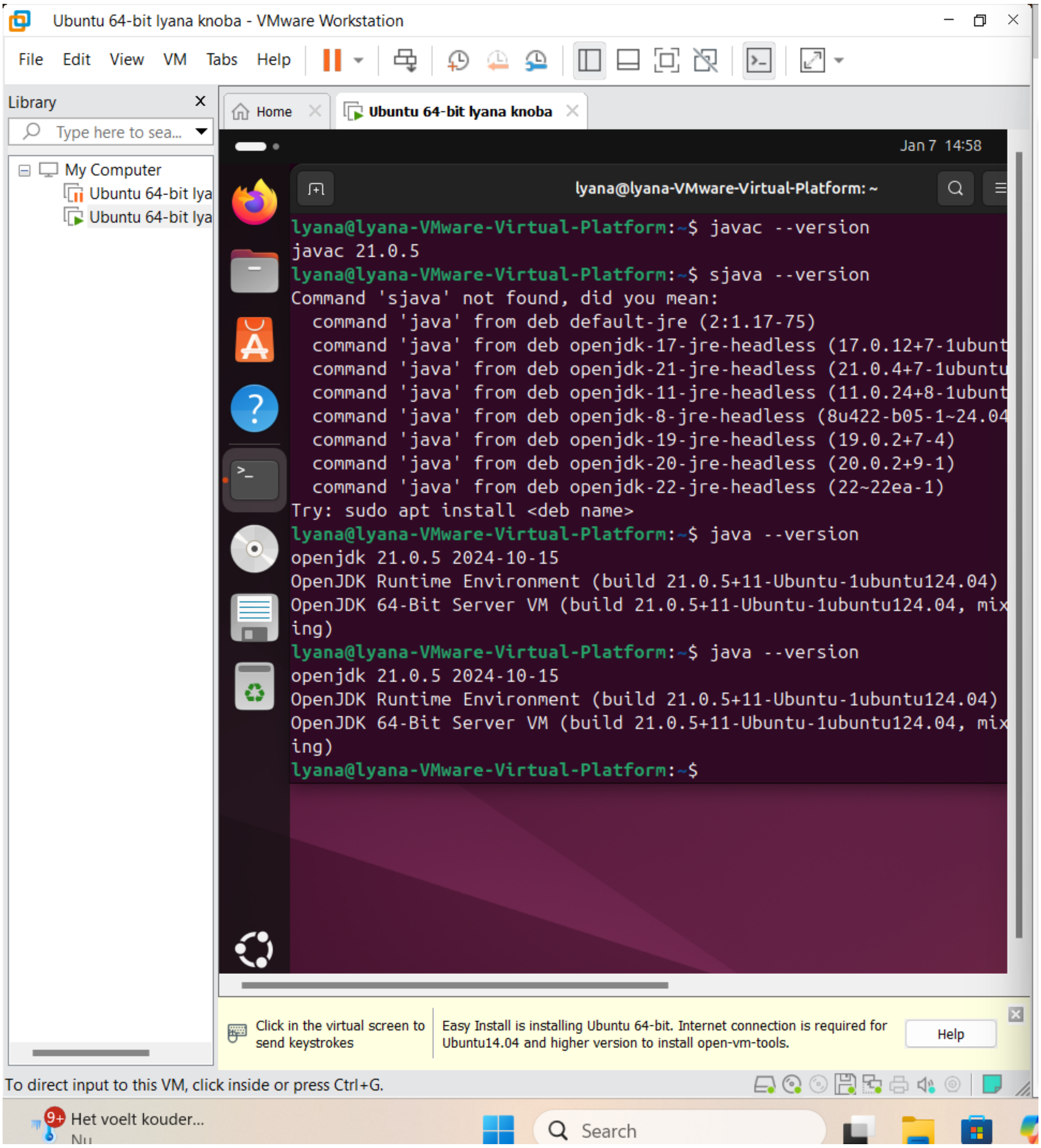
Assignment 4.2: Programming languages

Take screenshots that the following commands work:

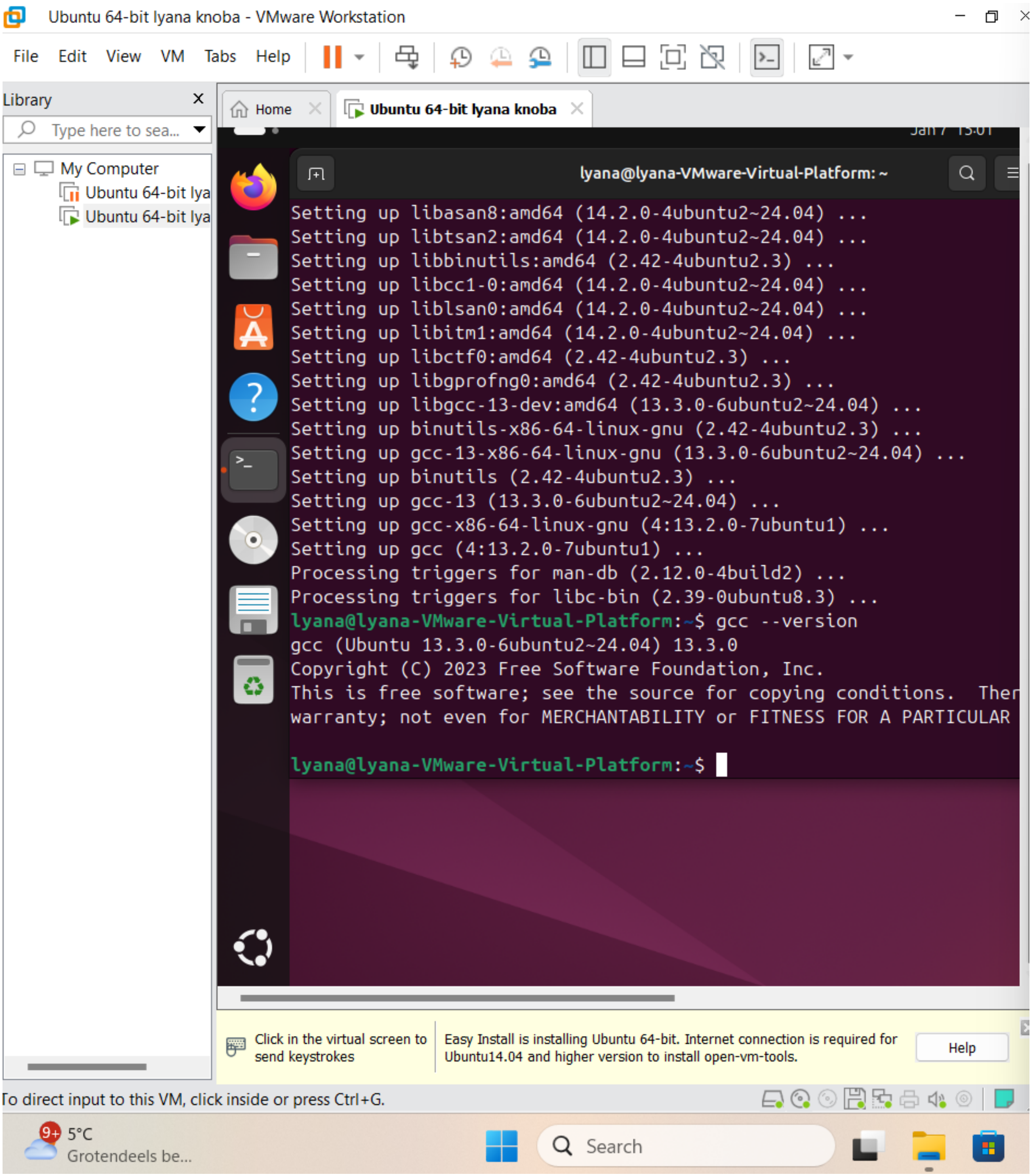
javac –version



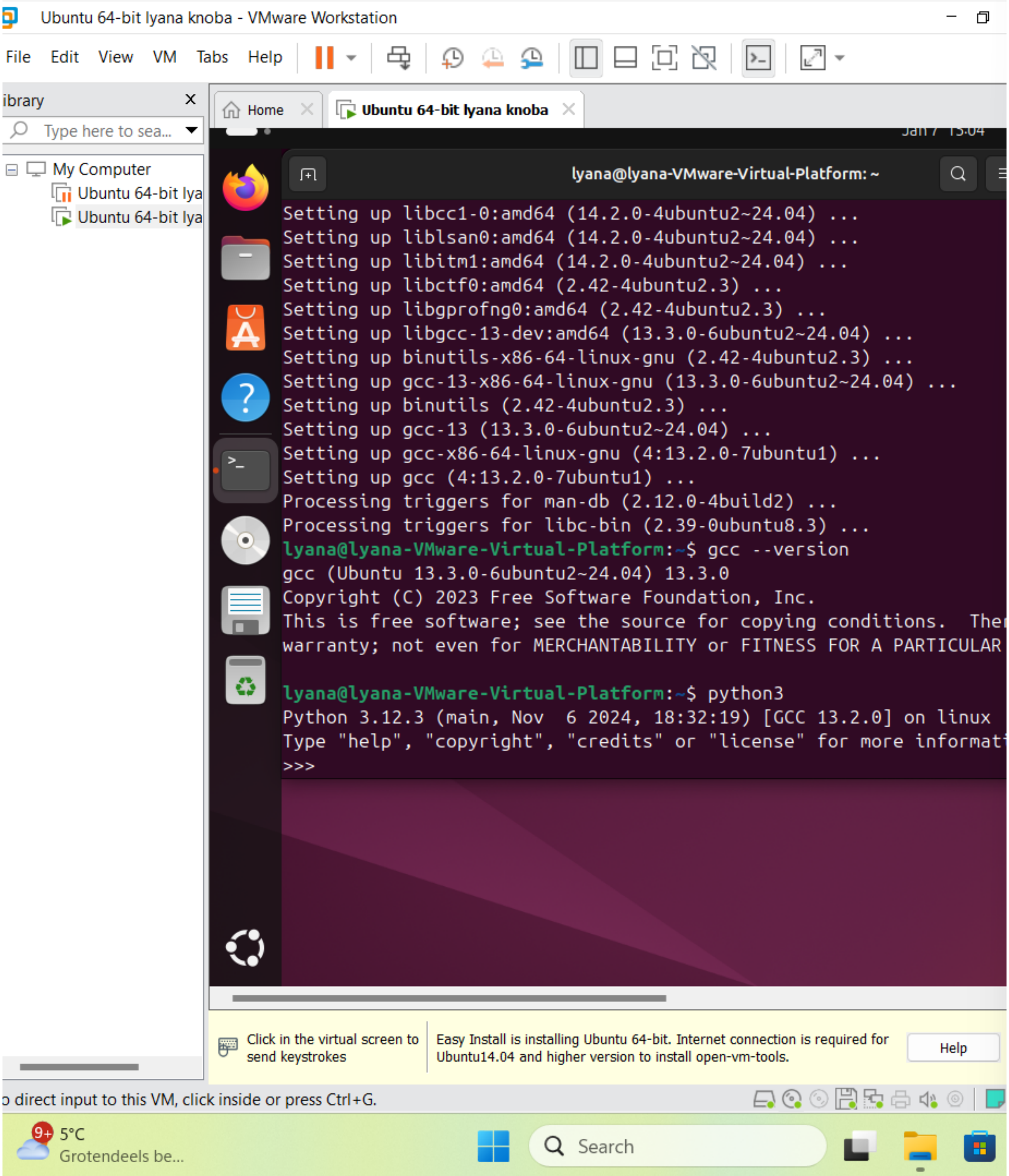
java –version


Ubuntu 64-bit lyana knoba - VMware Workstation
File Edit View VM Tabs Help
Library
Type here to sea...
My Computer
Ubuntu 64-bit lya
Ubuntu 64-bit lya
Home
Ubuntu 64-bit lyana knoba
Jan 7 14:58
lyana@lyana-VMware-Virtual-Platform: ~
lyana@lyana-VMware-Virtual-Platform:~$ javac --version
javac 21.0.5
lyana@lyana-VMware-Virtual-Platform:~$ sjava --version
Command 'sjava' not found, did you mean:
command 'java' from deb default-jre (2:1.17-75)
command 'java' from deb openjdk-17-jre-headless (17.0.12+7-1ubunt
command 'java' from deb openjdk-21-jre-headless (21.0.4+7-1ubuntu
command 'java' from deb openjdk-11-jre-headless (11.0.24+8-1ubunt
command 'java' from deb openjdk-8-jre-headless (8u422-b05-1~24.04
command 'java' from deb openjdk-19-jre-headless (19.0.2+7-4)
command 'java' from deb openjdk-20-jre-headless (20.0.2+9-1)
command 'java' from deb openjdk-22-jre-headless (22~22ea-1)
Try: sudo apt install <deb name>
lyana@lyana-VMware-Virtual-Platform:~$ java --version
openjdk 21.0.5 2024-10-15
OpenJDK Runtime Environment (build 21.0.5+11-Ubuntu-1ubuntu124.04)
OpenJDK 64-Bit Server VM (build 21.0.5+11-Ubuntu-1ubuntu124.04, mix
ing)
lyana@lyana-VMware-Virtual-Platform:~$ java --version
openjdk 21.0.5 2024-10-15
OpenJDK Runtime Environment (build 21.0.5+11-Ubuntu-1ubuntu124.04)
OpenJDK 64-Bit Server VM (build 21.0.5+11-Ubuntu-1ubuntu124.04, mix
ing)
lyana@lyana-VMware-Virtual-Platform:~$
Click in the virtual screen to send keystrokes
Easy Install is installing Ubuntu 64-bit. Internet connection is required for Ubuntu14.04 and higher version to install open-vm-tools.
Help
To direct input to this VM, click inside or press Ctrl+G.
Het voelt kouder...
Search

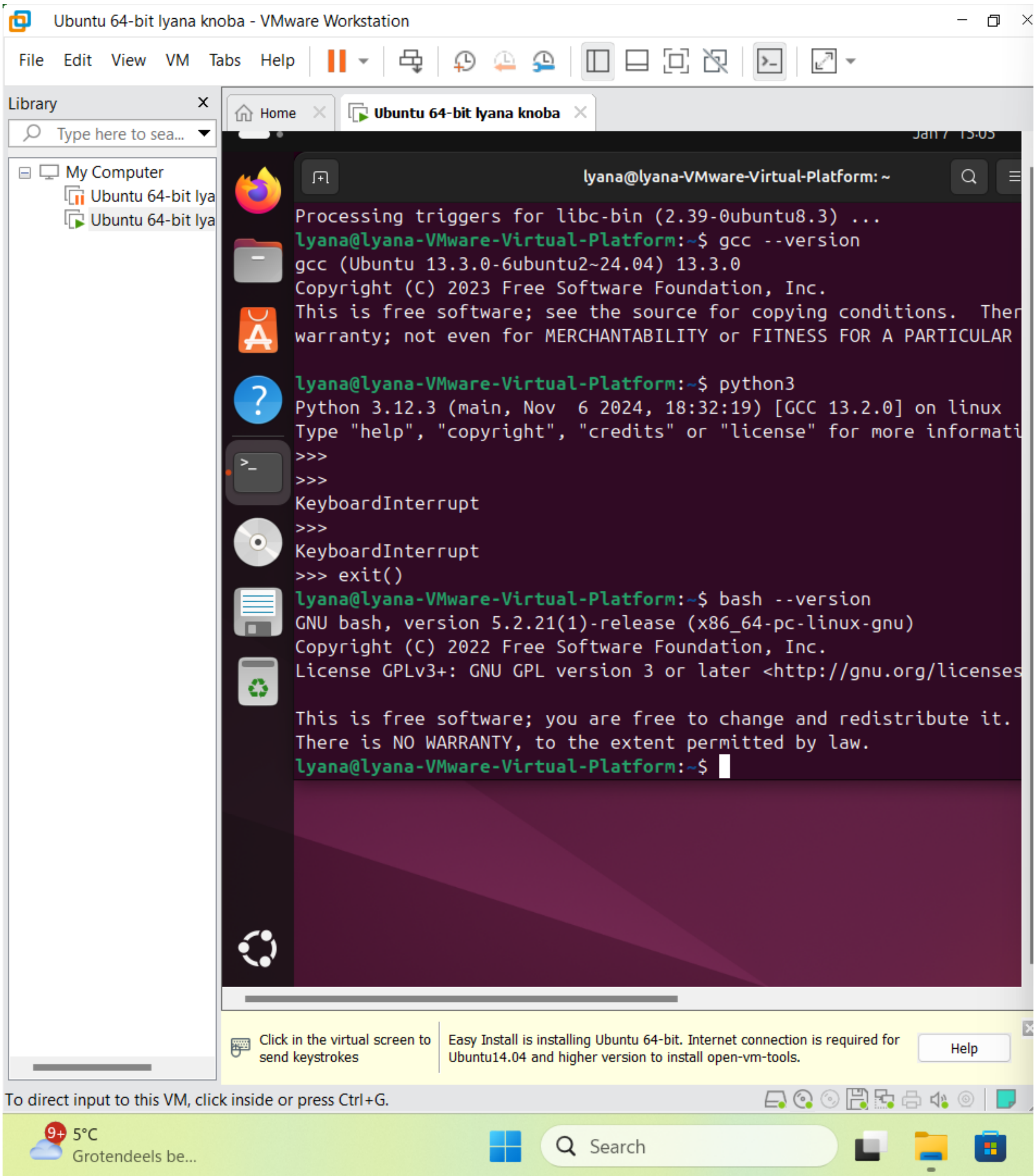
gcc –version



python3 –version



bash –version


Ubuntu 64-bit lyana knoba - VMware Workstation
File Edit View VM Tabs Help
Library
Type here to sea...
My Computer
Ubuntu 64-bit lya
Ubuntu 64-bit lya
Home
Ubuntu 64-bit lyana knoba
lyana@lyana-VMware-Virtual-Platform: ~
Processing triggers for libc-bin (2.39-0ubuntu8.3) ...
lyana@lyana-VMware-Virtual-Platform:~$ gcc --version
gcc (Ubuntu 13.3.0-6ubuntu2~24.04) 13.3.0
Copyright (C) 2023 Free Software Foundation, Inc.
This is free software; see the source for copying conditions. Ther
warranty; not even for MERCHANTABILITY or FITNESS FOR A PARTICULAR
lyana@lyana-VMware-Virtual-Platform:~$ python3
Python 3.12.3 (main, Nov 6 2024, 18:32:19) [GCC 13.2.0] on linux
Type "help", "copyright", "credits" or "license" for more informati
>>>
>>>
KeyboardInterrupt
>>>
KeyboardInterrupt
>>> exit()
lyana@lyana-VMware-Virtual-Platform:~$ bash --version
GNU bash, version 5.2.21(1)-release (x86_64-pc-linux-gnu)
Copyright (C) 2022 Free Software Foundation, Inc.
License GPLv3+: GNU GPL version 3 or later <http://gnu.org/licenses
This is free software; you are free to change and redistribute it.
There is NO WARRANTY, to the extent permitted by law.
lyana@lyana-VMware-Virtual-Platform:~$
Click in the virtual screen to send keystrokes
Easy Install is installing Ubuntu 64-bit. Internet connection is required for Ubuntu14.04 and higher version to install open-vm-tools.
Help
To direct input to this VM, click inside or press Ctrl+G.
5°C
Grotendeels be...
Search

Assignment 4.3: Compile

Which of the above files need to be compiled before you can run them?: C and Java files need to be compiled before running. So, fib.c needs to be compiled with a C compiler and Fibonacci.java needs to be compiled with the java compiler.

Which source code files are compiled into machine code and then directly executable by a processor?: The C source file is compiled into machine code and can be executed directly by the processor.

Which source code files are compiled to byte code?: The java source file is compiled into byte code.

Which source code files are interpreted by an interpreter?: The python source file is interpreted by the python interpreter and the bash script is interpreted by the shell.

These source code files will perform the same calculation after compilation/interpretation. Which one is expected to do the calculation the fastest?: The C program performs the calculation the fastest because it is compiled into native machine code and can directly execute on the processor.

How do I run a Java program?: first you need to compile the source code using the javac compiler by typing “javac Fibonacci.java” in the terminal. Then run the command by typing “java Fibonacci”

How do I run a Python program?: type in “python3 fib.py”

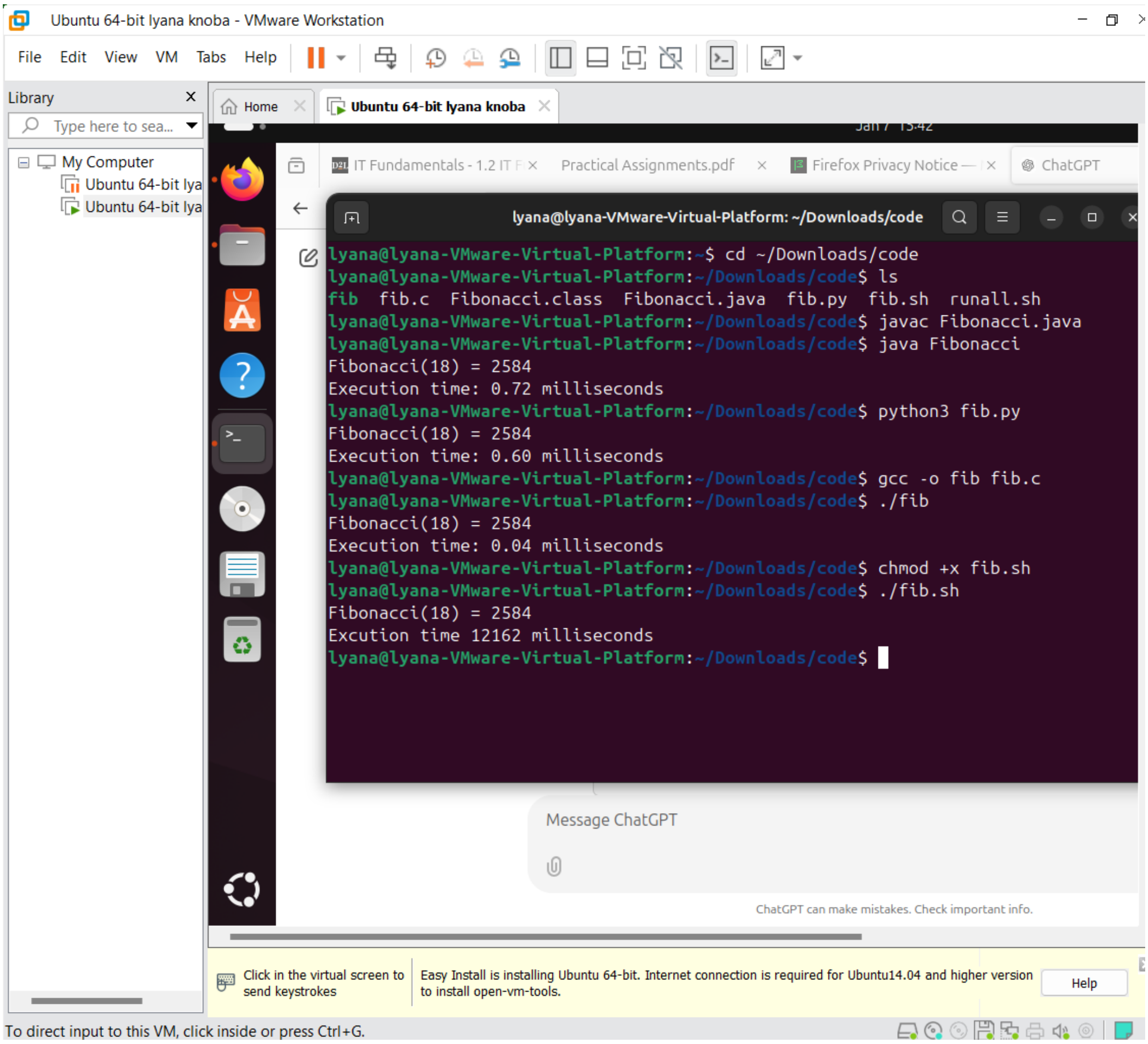
How do I run a C program?: first you need to compile it by typing in “gcc -o fib fib.c” and run it by typing in “./fib”

How do I run a Bash script?: first make the script executable by typing in “chmod +x fib.sh” and then run it by typing in “./fib.sh”

If I compile the above source code, will a new file be created? If so, which file?: yes, new files will be created for fib.c and Fibonacci.java. the python and bash scripts don’t need compilation so no new files are created.

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?

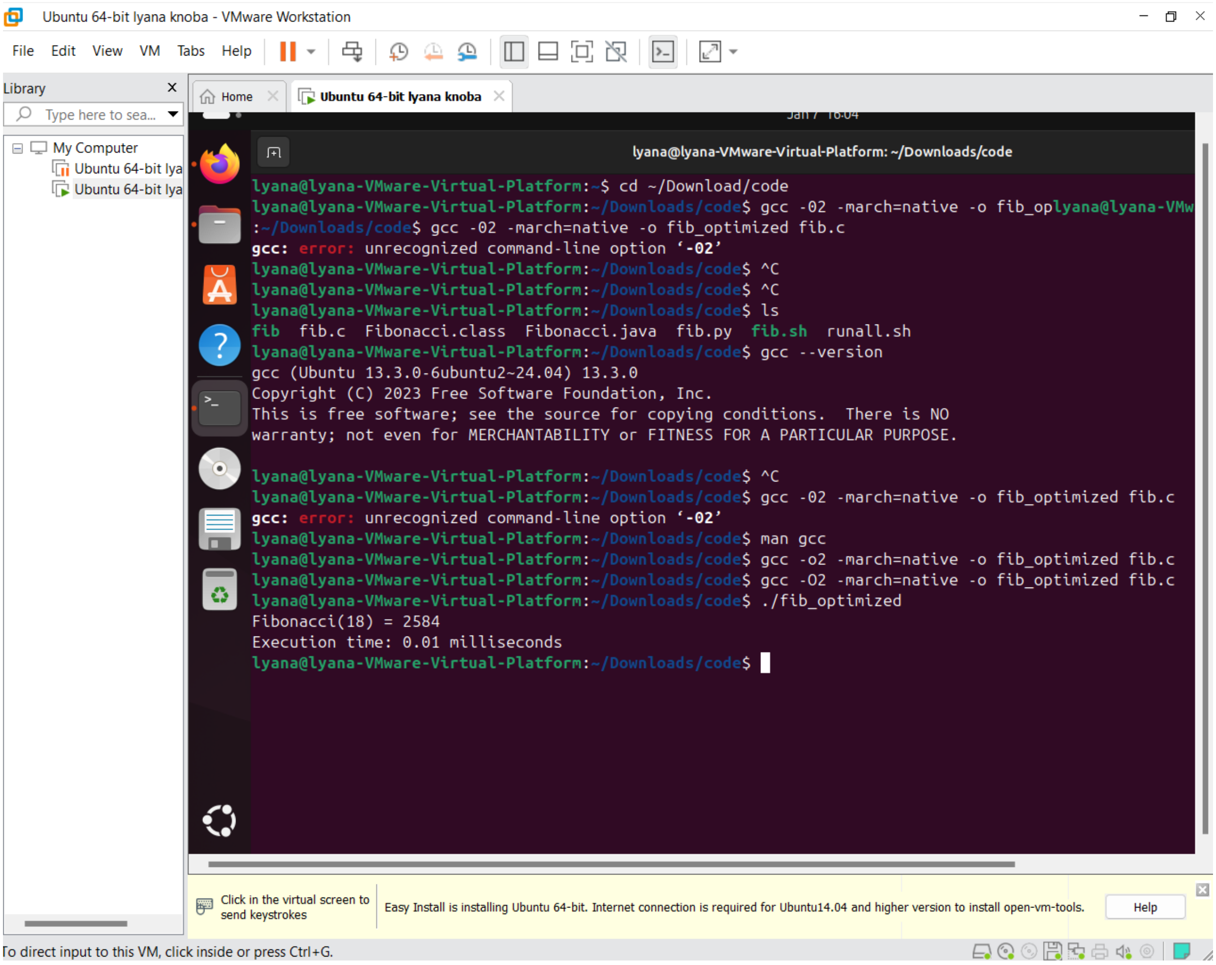


The c source code runs the fastest. It only took 0.04 milliseconds while the bash script took the longest at 12162 milliseconds.

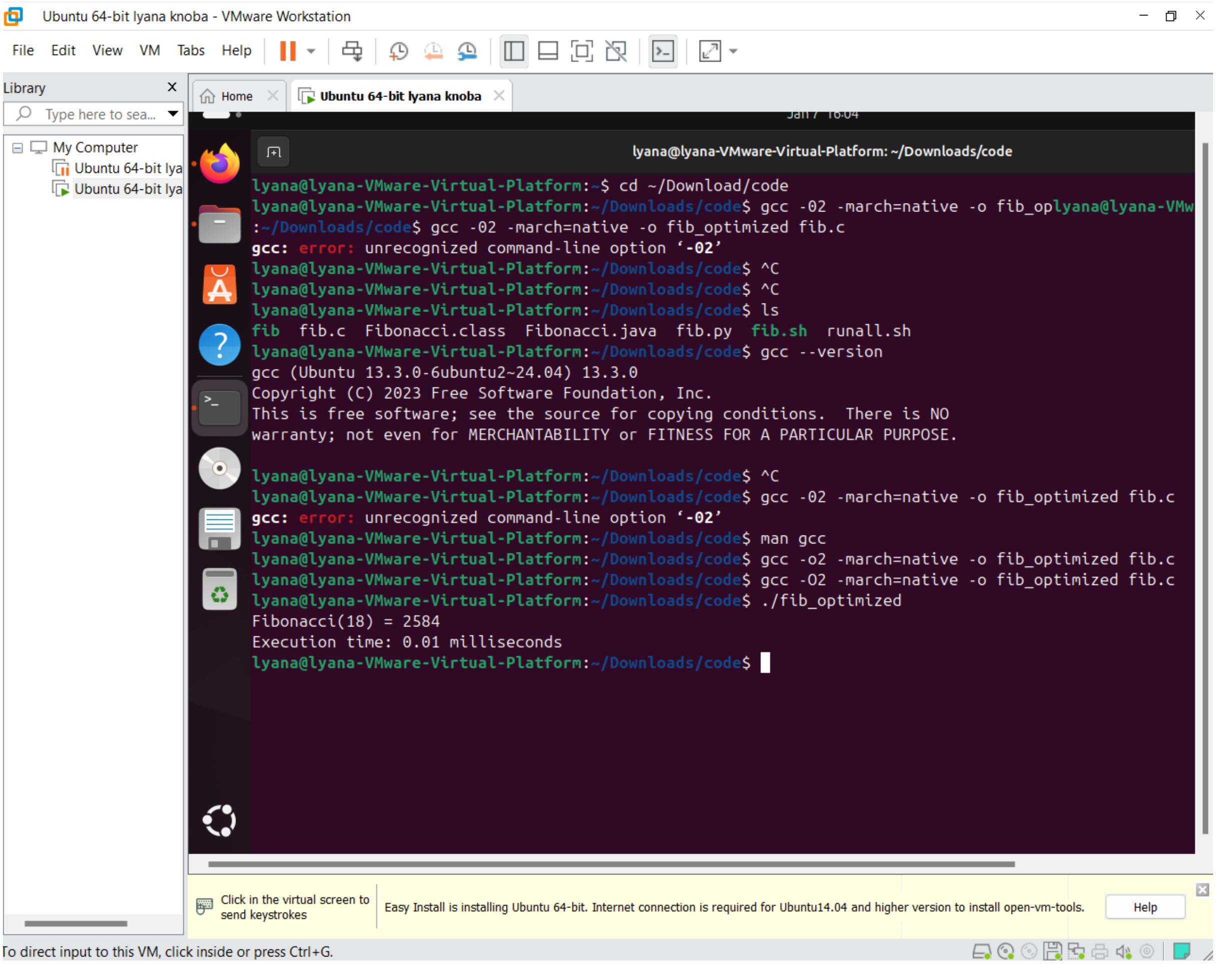
Assignment 4.4: Optimize

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.

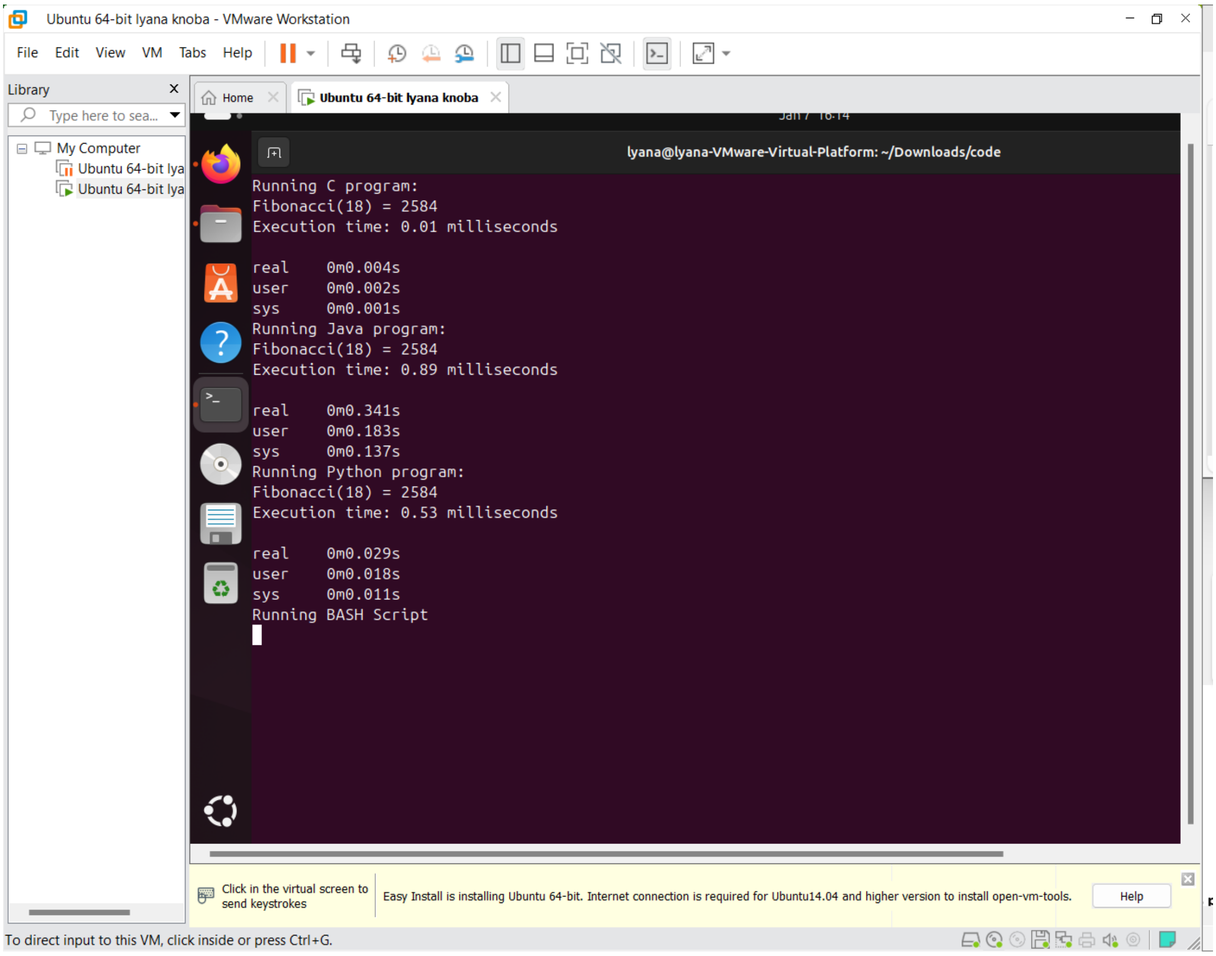


b) Compile **fib.c** again with the optimization parameters



c) Run the newly compiled program. Is it true that it now performs the calculation faster?
Yes, it is true. It's execution time is now 0.01 milliseconds.

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) 

Bonus point assignment – week 4

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.

```
_start:
    mov r1, #2
    mov r2, #4
    mov r0, #1

Loop:
    mul r0, r0, r1
    subs r2, r2, #1
    bne Loop
```

End:

(Note to self, in case I come back and have no clue about what is going on, here, we are storing in r0 and we have r2 which has the value of 4 as the counter. In the loop r1(2)*r0(1) = r0(2) then r2 decreases by 1 which now is 3. Now r0(2)*r1(2) = r0(4) now r2 decreases to 2 and this loops until r2 becomes 0

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**