PROGRAMMAZIONE PROCEDURALE

A.A. 2023/2024

WHAT YOU NEED

- The good news for anyone using **Linux** or **Mac OSX** is that you are on a system designed *for* programming in C. The authors of the C language were also instrumental in the creation of the Unix operating system, and both Linux and OSX are based on Unix. In fact, the install will be incredibly easy.
- You need

✓**A shell (bash)**

✓**A compiler: we will use GCC**

  - try typing “gcc --version” in your shell

✓**A code editor**

SHELL (BASH)

- Bash is a Unix shell and command language written by Brian Fox for the GNU Project as a free software replacement for the Bourne shell. First released in 1989, it has been distributed widely as the default login shell for most Linux distributions and Apple's macOS (formerly OS X).
- Bash is a command processor that typically runs in a text window, where the user types commands that cause actions.
- Howto in italian:
  - ✓https://www.gnu.org/software/bash/manual/bash.html
- Slides on my page (we only need cd, ls and a few others)

GCC

- https://gcc.gnu.org
- GNU C Compiler
- The GNU Compiler Collection includes front ends for C, C++, Objective-C, Fortran, Java, Ada, and Go
- GCC 11.2 last release [28th July 2021]
  - ✓Manual https://gcc.gnu.org/onlinedocs/gcc-11.2.0/gcc.pdf
  - ✓Also an older version is ok

LINUX

- On most Linux systems you just have to install a few packages. For Debian based systems, like Ubuntu you should just have to install a few things using these commands:

  ```
  $ sudo apt install gcc
  ```

- The above is an example of a command line prompt, so to get to where you can run that, find your "Terminal" program and run it first. Then you'll get a shell prompt similar to the $ above and can type that command into it.

- Here's how you would install the same setup on an RPM based Linux like Fedora:

  ```
  $ su -c "yum groupinstall development-tools"
  ```

- Once you've run that, you should be able to do the first Exercise in this book and it'll work. If not then let me know.

MAC OSX

- On Mac OSX the installation is even easier. First, you'll need to either download the latest XCode from Apple, or find your install DVD and install it from there.

WINDOWS

- https://www.cygwin.com
- Cygwin is:
  - ✓a large collection of GNU and Open Source tools which provide functionality similar to a Linux distribution on Windows.
  - ✓a DLL (cygwin1.dll) which provides substantial POSIX API functionality.

SUGGESTIONS AND HOW TO CYGWIN

- First suggestion is: do not use CYGWIN ☺
  - ✓Install Linux on you laptop/computer
- If you really want to use Cygwin
  - ✓Remember to select among Devel packages **cygwin64-ggc-core**
  - ✓Once installed click on **cygwin.bat** to launch the shell
  - ✓Then try to digit “gcc”, if it says it exists, OK!
  - ✓If it dos not exist, maybe it has a weird name like i686-pc-cygwin-gcc.exe. In the shell type the following commands that link the name “gcc” to the name of the compiler
    - cd /usr/bin/
    - ln i686-pc-cygwin-gcc.exe gcc
- https://www3.ntu.edu.sg/home/ehchua/programming/howto/Cygwin_HowTo.html

IF YOU HAVE WINDOWS

- Install Linux (Ubuntu) on a different partition (more complicated)

- **BETTER**: use a virtual machine
- Oracle VM VirtualBox is a **free** and open-source hosted hypervisor for x86 virtualization, developed by Oracle Corporation
  - ✓https://www.virtualbox.org/wiki/Downloads

- Suggested Linux: **Ubuntu 22.04.3 LTS**
  - ✓https://ubuntu.com/download/desktop

TUTORIAL

- Install Ubuntu on Windows Using VirtualBox
  - ✓https://www.debugpoint.com/install-ubuntu-windows-virtualbox/

MAYBE VMWARE IS BETTER

- Free VMwre Workstation player 17
  - ✓https://www.vmware.com/go/getplayer-win

- How to Install Ubuntu on Windows using VMware?
  - ✓https://www.geeksforgeeks.org/how-to-install-ubuntu-on-windows-using-vmware/

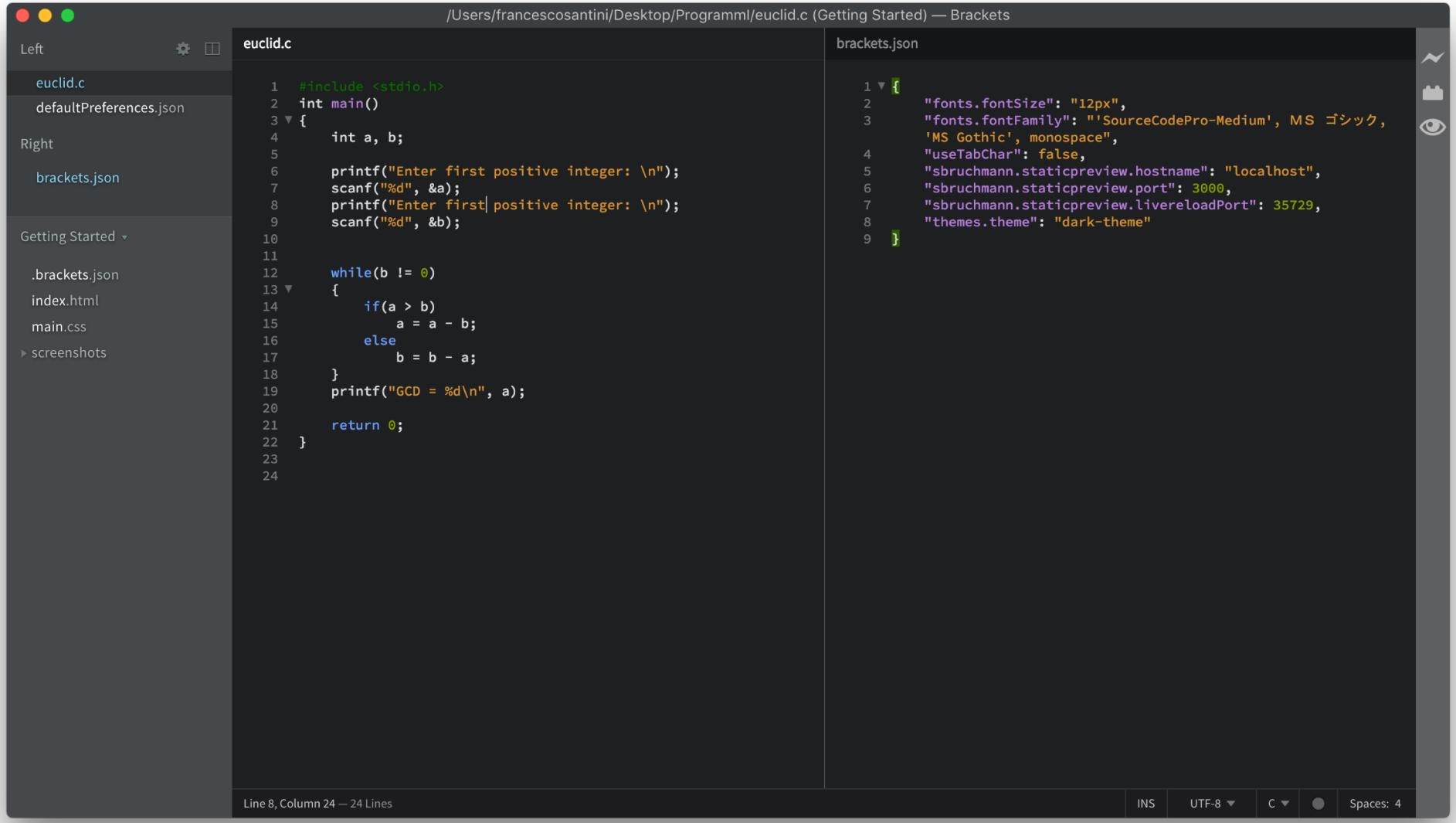
TEXT EDITOR

- The choice of text editor for a programmer is a tough one. For beginners I tell them to just use Gedit since it's simple and works for code. However, it doesn't work in certain internationalized situations, and chances are you already have a favorite text editor if you've been programming for a while.
  - ✓**Atom** on Windows, Linux, MacOS
  - ✓**Visual Studio Code**, Windows, Linux, MacOS
  - ✓Brackets on Windows, Linux, MacOS
  - ✓Gedit on Linux and MacOS
  - ✓Emacs and Emacs for OSX. Be prepared to do some learning (in CC mode: mode for editing C, C++, Objective C, Java, Pike, and IDL code.)
  - ✓Vim and MacVim

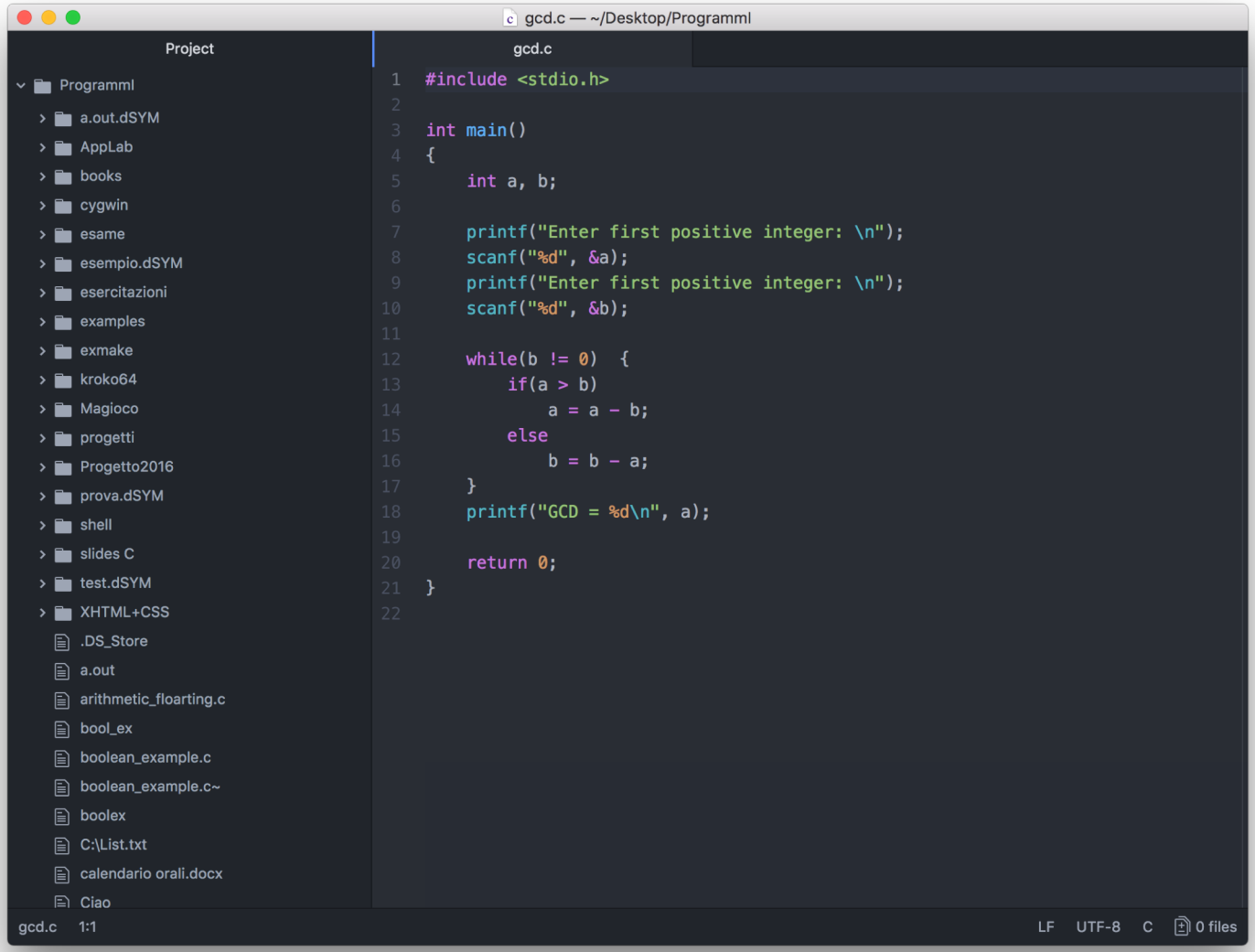
LINKS

- Atom https://atom.io (suggested)
- Brackets http://brackets.io
- Gedit https://wiki.gnome.org/Apps/Gedit
- GNU Emacs https://www.gnu.org/software/emacs/
  - ✓CC mode https://www.gnu.org/software/emacs/manual/html_mono/ccmode.html
  - ✓When you open a .c file it automatically goes into CC mode
- Visual Studio Code (https://visualstudio.microsoft.com)

BRACKETS

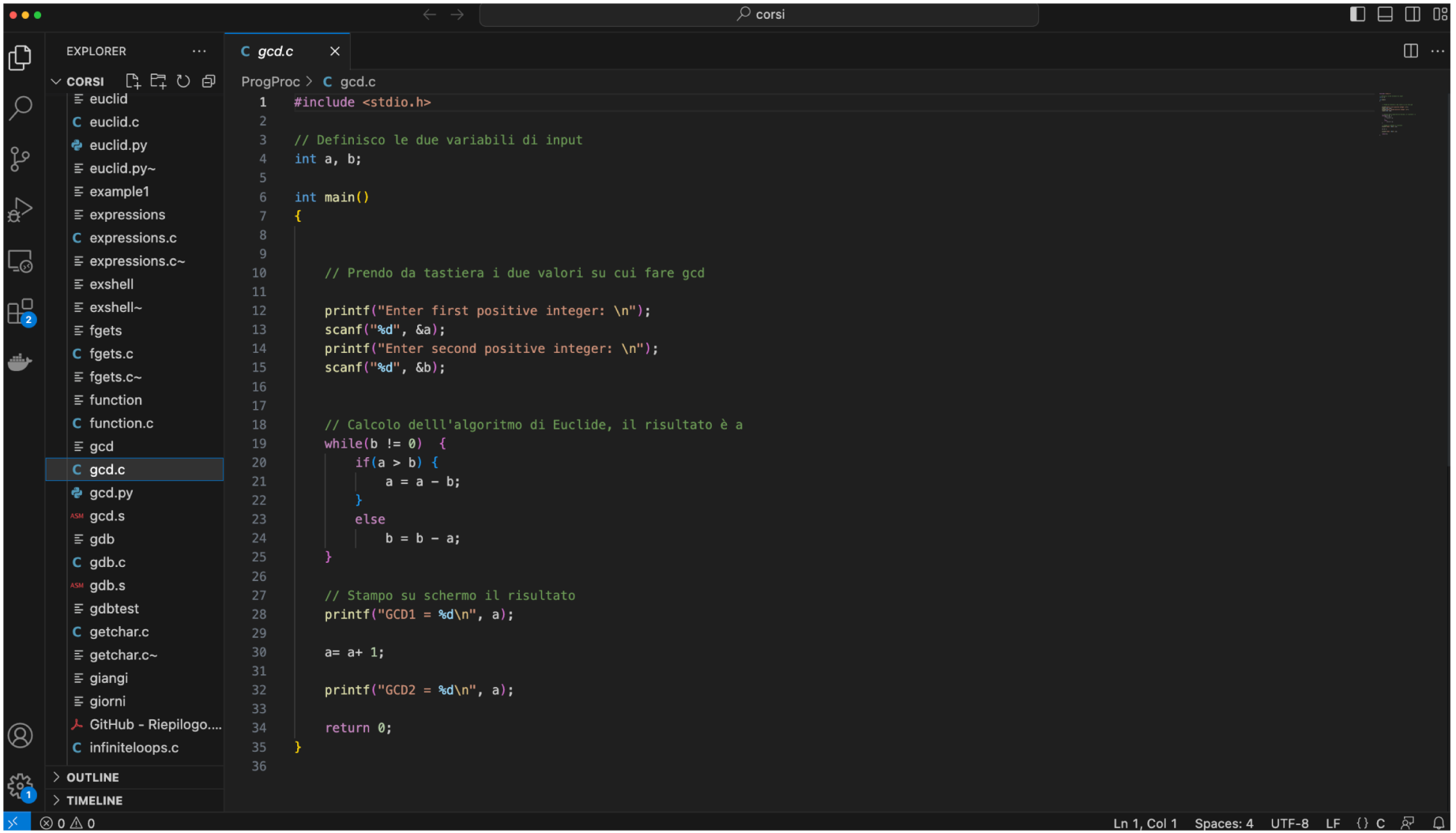

/Users/francescosantini/Desktop/Programml/euclid.c (Getting Started) — Brackets
Left
euclid.c
defaultPreferences.json
Right
brackets.json
Getting Started
.brackets.json
index.html
main.css
screenshots
euclid.c
#include <stdio.h>
int main()
{
int a, b;
printf("Enter first positive integer: \n");
scanf("%d", &a);
printf("Enter first positive integer: \n");
scanf("%d", &b);
while(b != 0)
{
if(a > b)
a = a - b;
else
b = b - a;
}
printf("GCD = %d\n", a);
return 0;
}
brackets.json
{
"fonts.fontSize": "12px",
"fonts.fontFamily": "'SourceCodePro-Medium', MS ゴシック, 'MS Gothic', monospace",
"useTabChar": false,
"sbruchmann.staticpreview.hostname": "localhost",
"sbruchmann.staticpreview.port": 3000,
"sbruchmann.staticpreview.livereloadPort": 35729,
"themes.theme": "dark-theme"
}
Line 8, Column 24 — 24 Lines
INS
UTF-8
C
Spaces: 4

ATOM



NON COMPILARE DA DENTRO ATOM!

VISUAL STUDIO



NON COMPILARE DA DENTRO VISUAL STUDIO!
Ok se si compila da terminal dentro Visual Studio, con gcc

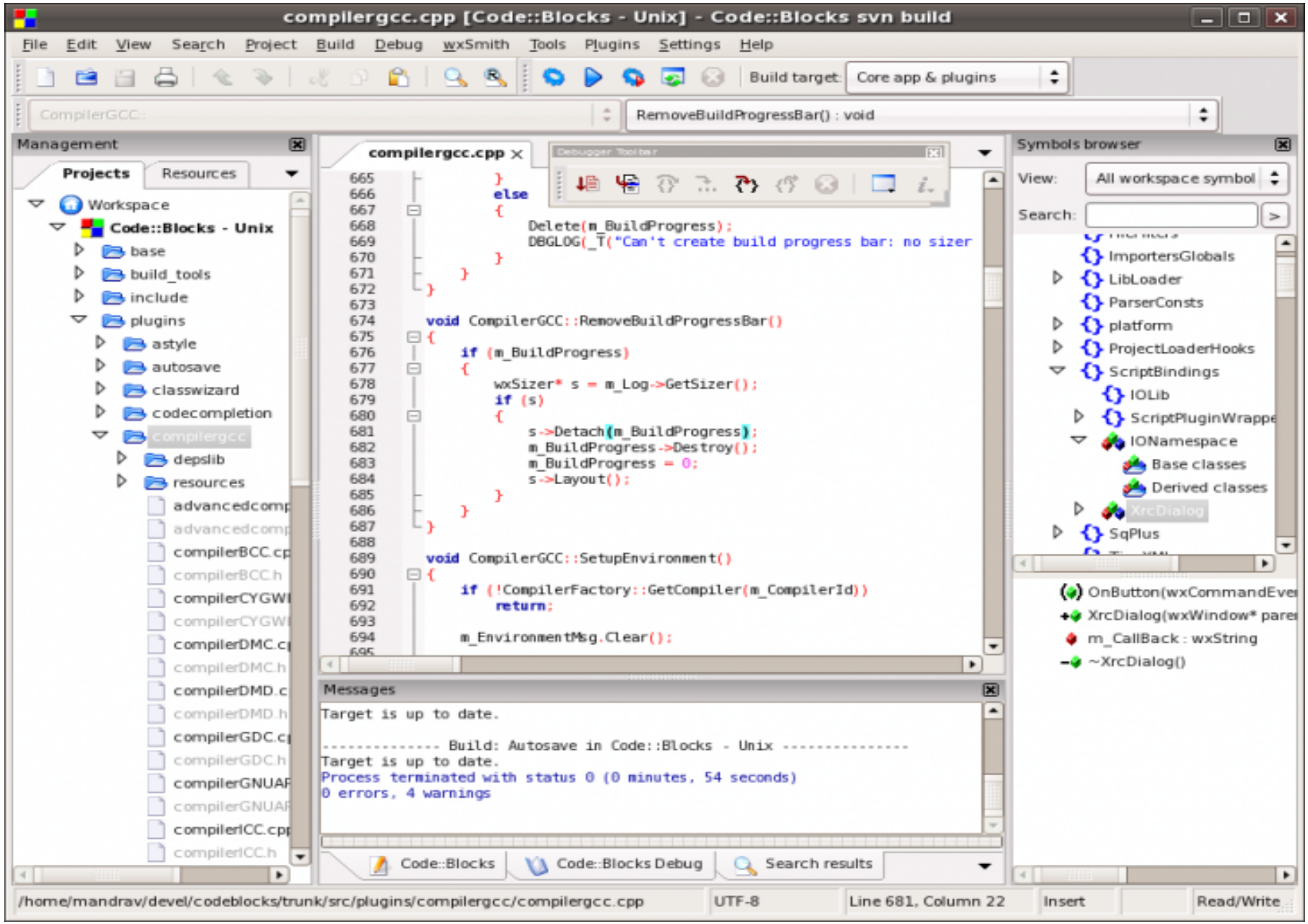
INSTALL ATOM / VS CODE

- Atom
  - ✓https://linuxize.com/post/how-to-install-atom-text-editor-on-ubuntu-20-04/
- Visual Studio Code
  - ✓https://linuxize.com/post/how-to-install-visual-studio-code-on-ubuntu-20-04/

WARNING

- An IDE, or "Integrated Development Environment" will turn you stupid. They are the worst tools if you want to be a good programmer because they hide what's going on from you, and your job is to know what's going on.
- They are useful if you're trying to get something done and the platform is designed around a particular IDE, but for learning to code C (and many other languages) they are pointless.
- We will enjoy being stupid at the end of the course: after you will be allowed to use an IDE

CODE::BLOCKS
HTTP://WWW.CODEBLOCKS.ORG



HELLO WORLD

1) Write and save it to a test.c file

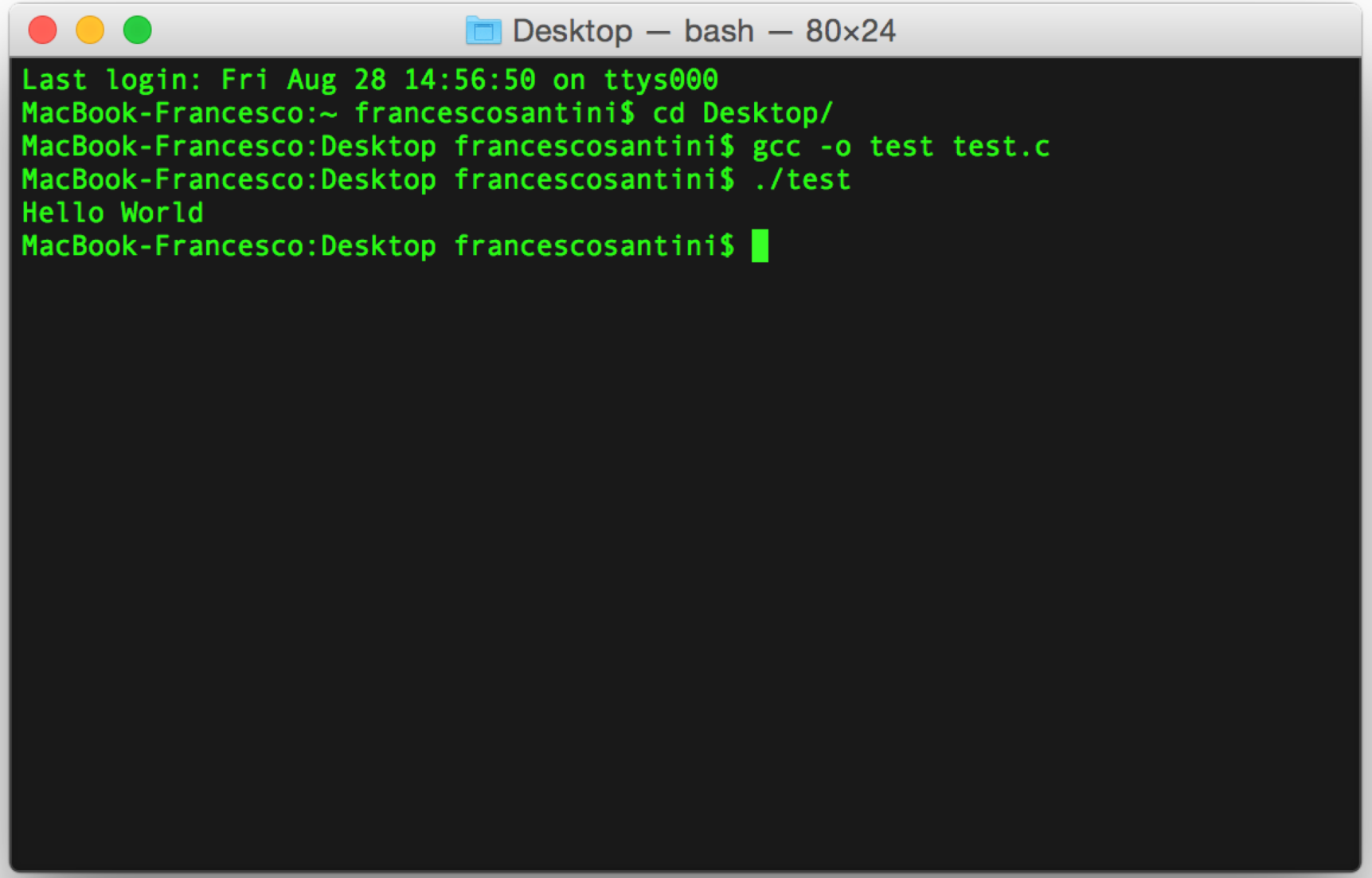
```
#include <stdio.h>

int main ()
{
    printf("Hello World\n");
    return 0;
}
```

2) Compile it: gcc -o test test.c

3) Then execute it: ./test

FROM BASH IN MAC OSX



PLEASE BE SURE TO BE ABLE TO WRITE
CODE AND COMPILE IT
AS SOON AS POSSIBLE !!!!!!!!!!

ATTENZIONE

- **NON** correggerò nessun progetto sviluppato tramite Xcode, Code::Blocks, Netbeans, Eclipse
- Allora prova di lavoratorio dovete saper compilare solo con gcc

SOLAMENTE GCC

EXAMPLE 1

```
Enter two integers: 12
11
12 + 11 = 23
```

```c
#include <stdio.h>
int main()
{
    int firstNumber, secondNumber, sumOfTwoNumbers;

    printf("Enter two integers: ");

    // Two integers entered by user is stored using scanf()
function
    scanf("%d %d", &firstNumber, &secondNumber);

    // sum of two numbers in stored in variable sumOfTwoNumbers
    sumOfTwoNumbers = firstNumber + secondNumber;

    // Displays sum
    printf("%d + %d = %d", firstNumber, secondNumber,
sumOfTwoNumbers);

    return 0;
}
```

EXAMPLE 2

```c
#include <stdio.h>
int main(){

    int dividend, divisor, quotient, remainder;

    printf("Enter dividend: ");
    scanf("%d", &dividend);

    printf("Enter divisor: ");
    scanf("%d", &divisor);

    // Computes quotient
    quotient = dividend/divisor;

    // Computes remainder
    remainder = dividend%divisor;

    printf("Quotient = %d\n",quotient);
    printf("Remainder = %d",remainder);

    return 0;
}
```

```
Enter dividend: 25
Enter divisor: 4
Quotient = 6
Remainder = 1
```

EXAMPLE 3

```
#include <stdio.h>
int main()
{
    int n1, n2;

    printf("Enter two positive integers: ");
    scanf("%d %d",&n1,&n2);

    while(n2!=0)
    {
        if(n1 > n2)
            n1 -= n2;
        else
            n2 -= n1;
    }
    printf("GCD = %d\n",n1);

    return 0;
}
```

```
Enter two positive integers: 81
153
GCD = 9
```

SU LIBRO

- Sezioni 1.10 con particolare attenzione alla 1.10.3
  - ✓No 1.10.4
- Sezione 1.8