# 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 <u>for</u> 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.
- Q 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.
- e Howto in italian:

✓ <u>https://www.gnu.org/software/bash/manual/bash.html</u>

Ilides 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 [28<sup>th</sup> July 2021]
  - Manual <u>https://gcc.gnu.org/onlinedocs/gcc-11.2.0/gcc.pdf</u>
  - ✓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.
- e 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
- Over the constraint of the
  - ✓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 ③
  - ✓ 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-cygwingcc.exe. In the shell type the following commands that link the name "gcc" to the name of the compiler
    - cd /usr/bin/
    - In i686-pc-cygwin-gcc.exe gcc
- https://www3.ntu.edu.sg/home/ehchua/programming/howto/C ygwin\_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

 ✓ <u>https://www.debugpoint.com/install-ubuntu-windows-</u> virtualbox/

## **MAYBE VMWARE IS BETTER**

Free VMwre Workstation player 17

✓ <u>https://www.vmware.com/go/getplayer-win</u>

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



# **TEXT EDITOR**

- Provide the second structure of the second structur
  - ✓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 <u>https://atom.io</u> (suggested)
- Ø Brackets <u>http://brackets.io</u>
- Gedit <u>https://wiki.gnome.org/Apps/Gedit</u>
- GNU Emacs https://www.gnu.org/software/emacs/
  - ✓CC mode
    - https://www.gnu.org/software/emacs/manual/html\_mono/ccm ode.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	brackets.json			
euclid.c defaultPreferences.json Right brackets.json Getting Started ↓ .brackets.json index.html main.css b screenshots	<pre>intlude (stdio.b) int main() {     int a, b;     printf("Enter first positive integer: \n");     scanf("%d", &amp;a);     printf("Enter first] positive integer: \n");     scanf("%d", &amp;b);      while(b != 0)</pre>	<pre>1 v { 2  "fonts.fontSize": "12px", 3  "fonts.fontFamily": "'SourceCodePro-Medium', MS I&gt;vy6,     'NsGabChar": false, 5  "sbruchmann.staticpreview.hostname": "localhost", 6  "sbruchmann.staticpreview.port": 3000, 7  struchmann.staticpreview.livereloadPort": 35729, 8  "themes.theme": "dark-theme" 9  ]</pre>			
	Line 8, Column 24 — 24 Lines	INS UTF-8 🕶 C 💌 💿 Spaces: 4			

#### NON COMPILARE DA DENTRO ATOM!





## **VISUAL STUDIO**

•••		$\leftarrow  ightarrow$ ( $ ho$ corsi	
பு	EXPLORER ····	C gcd.c ×	□ …
	∨ CORSI [°+ ⊑7, ບ໋ອ	ProgProc > C gcd.c	
Q	≣ euclid	1 #include <stdio.h></stdio.h>	
1	C euclid.c		
မှ	🗬 euclid.py	3 // Definisco le due variabili di input	
	≣ euclid.py~	-+ <b>Inca, b,</b>	
~	≡ example1	6 int main()	
±,>	≡ expressions	7 {	
	C expressions.c		
	≣ expressions.c~	9 10 // Decende de tections i due volcari eu qui fare red	
	≣ exshell	10 // Prendo da tastiera i due vatori su cui fare gco	
	≣ exshell~	<pre>12 printf("Enter first positive integer: \n");</pre>	
	≣ fgets	13 scanf("%d", &a);	
حش	C fgets.c	<pre>14 printf("Enter second positive integer: \n");</pre>	
	≣ fgets.c~	15 scant("%d", &b);	
		10	
	C function.c	18 // Calcolo delll'algoritmo di Euclide, il risultato è a	
	≣ gcd	19 while(b != 0) {	
	C gcd.c	20 if(a > b) {	
	🕏 gcd.py	21   a = a - b;	
	ASM gcd.s	22 } 23 else	
	≣ gdb	24 <b>b</b> = b - a;	
	C gdb.c		
	ASM gdb.s	26	
	≣ gdbtest	2/ // Stampo su schermo il risultato 28 printf("GCD1 = &d\n" a):	
	C getchar.c	29	
	≣ getchar.c~	30 a= a+ 1;	
	≣ giangi	31	
	≣ giorni	32 printf("GCD2 = %d\n", a);	
8	GitHub - Riepilogo	34 return Ø:	
	C infiniteloops.c	35 }	
\$201 ×			
<u> </u>			
<u>&gt;` </u>			Ln I, Col I Spaces: 4 UIF-8 LF {} C 📯 🖵

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

# **INSTALL ATOM / VS CODE**

Q Atom

✓ <u>https://linuxize.com/post/how-to-install-atom-text-editor-on-ubuntu-20-04/</u>

Visual Studio Code

✓ <u>https://linuxize.com/post/how-to-install-visual-studio-code-on-ubuntu-20-04/</u>

### 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.
- Provide the second of the s
- 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

compilergcc.cpp [Code::Blocks - Unix] - Code::Blocks svn build 📃 🗆 🗙						
<u>File Edit View Search Project Build Debug w</u> xSmith <u>Tools Plugins Settings H</u> elp						
📔 🖻 🚔 🖕 🐁 🖓 🖓 😭 😫 🧕 🥵 🗍 😒 🖕 🦻 🦻 🧐 🐼 Build target: Core app & plugins 🗦						
CompilerGCC::						
Management 🕱	compilergcc.cpp x Debugger Tool tar	Symbols browser				
Projects Resources 🔻	665 - ) I III VA 30 3. 75 67 60 1 7 4	View: All workspace symbol 💲				
✓      Ó Workspace	666 else	Sound Comment				
🗢 🗧 Code::Blocks - Unix	668 Delete(n_BuildProgress);	Search:				
base	669 DBGLOG(_T("Can't create build progress bar: no sizer					
build tools	671 - }	b D Libl order				
D include	672 }	Clocode				
	673 674 woid CompilerGCC::RenoveRuildProgressRar()	ParserConsts				
pidgins	675	P D platform				
V 🛃 astyle	676 if (m_BuildProgress)	ProjectLoaderHooks				
P 🔁 autosave	677 - {	ScriptBindings				
P Classwizard	679 if (s)	IOLIB				
Codecompletion	680 🗄 📢	ScriptPluginWrappe				
🗢 🔁 compilergod	681 s->Detach(m_BuildProgress);					
depslib	683 m BuildProgress = 0:	Base classes				
P resources	684 s->Layout();					
advancedcomr	685 - }	Derived classes				
advanced.com	687	XrcUialog				
advancedcom,	688	P SqPlus				
compilerBCC.cp	689 void CompilerGCC::SetupEnvironment()					
compilerBCC.h	690 - {	(A) On Putton (www.Common.d.Euro				
compilerCYGWI	692 return;	( ) OnButton(wxcommanuever				
compilerCYGWI	693	+ ✓ XrcDialog(wxWindow* parel				
compilerDMC.c	694 m_EnvironmentMsg.Clear();	m_CallBack : wxString				
compilerDMC.h						
compilerDMD.c Messages						
compilerDMD.h Target is up to date.						
compilerGDC.cr						
compilerGDCh Target is up to date.						
compilerGNUAF						
0 errors, 4 warnings						
compiledCC cor						
compiler(CC b						
Code::Blocks V Code::Blocks Debug Search results						
/home/mandrav/devel/codeblocks/trun	k/src/plugins/compilergcc/compilergcc.cpp UTF-8 Line 681, Column 22	Insert Read/Write				

### **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

Desktop — bash — 80×24 Last login: Fri Aug 28 14:56:50 on ttys000 MacBook-Francesco:~ francescosantini\$ cd Desktop/ MacBook-Francesco:Desktop francescosantini\$ gcc -o test test.c MacBook-Francesco:Desktop francescosantini\$ ./test Hello World MacBook-Francesco:Desktop francescosantini\$

### 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

```
#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**

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

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

```
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;
```

### **EXAMPLE 3**

```
#include <stdio.h>
                                        153
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