

PROGRAMMAZIONE PROCEDURALE

A.A. 2025/2026



INFO

- Ⓢ Programmazione Procedurale con Laboratorio: 73 ore (9 CFU, 7+2 laboratorio)
- Ⓢ Pagina Web corso:
<http://www.dmi.unipg.it/francesco.santini/progl.html>
- Ⓢ Pagina Unistudium: <https://www.unistudium.unipg.it>
- Ⓢ Orario Martedì 9:30-11:30 (A0), Mercoledì 8:30-10:30 (A2), Giovedì 8:30-11:30 (A0)
✓ <https://easyacademy.unipg.it/agendaweb/>
- Ⓢ Canali per comunicare (più lenti):
 - ✓ Ricevimento su appuntamento via email
 - ✓ francesco.santini@unipg.it

TELEGRAM (PIÙ VELOCE)

📧 Telegram: <https://t.me/safran>

PROGRAMMA

@ Programming in C

- ✓ Language Basics
- ✓ Types
- ✓ Literals
- ✓ Type conversions
- ✓ Expressions and operators
- ✓ Statements
- ✓ Loops
- ✓ Functions
- ✓ Arrays
- ✓ Pointers
- ✓ Structures, Unions
- ✓ Dynamic Memory Management
- ✓ Input and Output
- ✓ Pre-processing directives

Language



PROGRAMMA

- ④ General programming concepts
 - ✓ Interpreted and compiled languages
 - ✓ Scope
- ④ Programming in C
 - ✓ C Language
 - ✓ Compiling with GCC
 - ✓ Debugging C Programs with GDB
- ④ Linked Lists
 - ✓ Different representations
 - ✓ Common operations

OBIETTIVI APPRENDIMENTO

1. Problem solving
2. General programming languages concepts
3. C language
4. Use of a compiler and debugger
5. Linked Lists

Saper programmare (in C)

MATERIALE

- ④ Slide (su mia pagina Web e Unistudium), upload PRIMA della lezione
- ④ Libri (proposti successivamente)
- ④ Esercitazioni

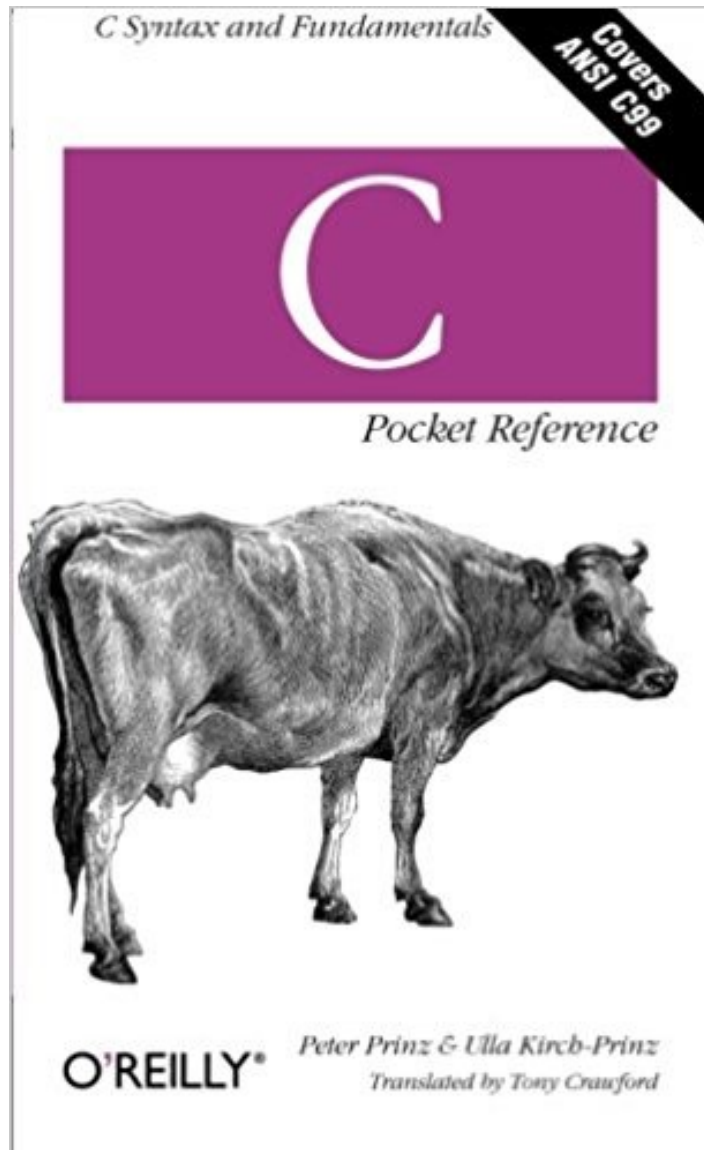
LIBRO PRINCIPALE



ITA

Il Linguaggio C – Fondamenti e tecniche di programmazione
By Paul J. Deitel and Harvey M. Deitel
Publisher: Pearson, 9th edition
(2022)
Pages: 736
Euro: 42,74 (Amazon)

RIFERIMENTO VELOCE



EN

C in a Nutshell (Pocket Reference)

By Peter Prinz, Ulla Kirch-Prinz

Publisher: O'Reilly Media

Final Release Date: November 2002 reprint
2022

Pages: 144

ALTRI LIBRI



ITA

Il linguaggio C. Principi di programmazione e manuale di riferimento

By B. Kernighan, D. Ritchie

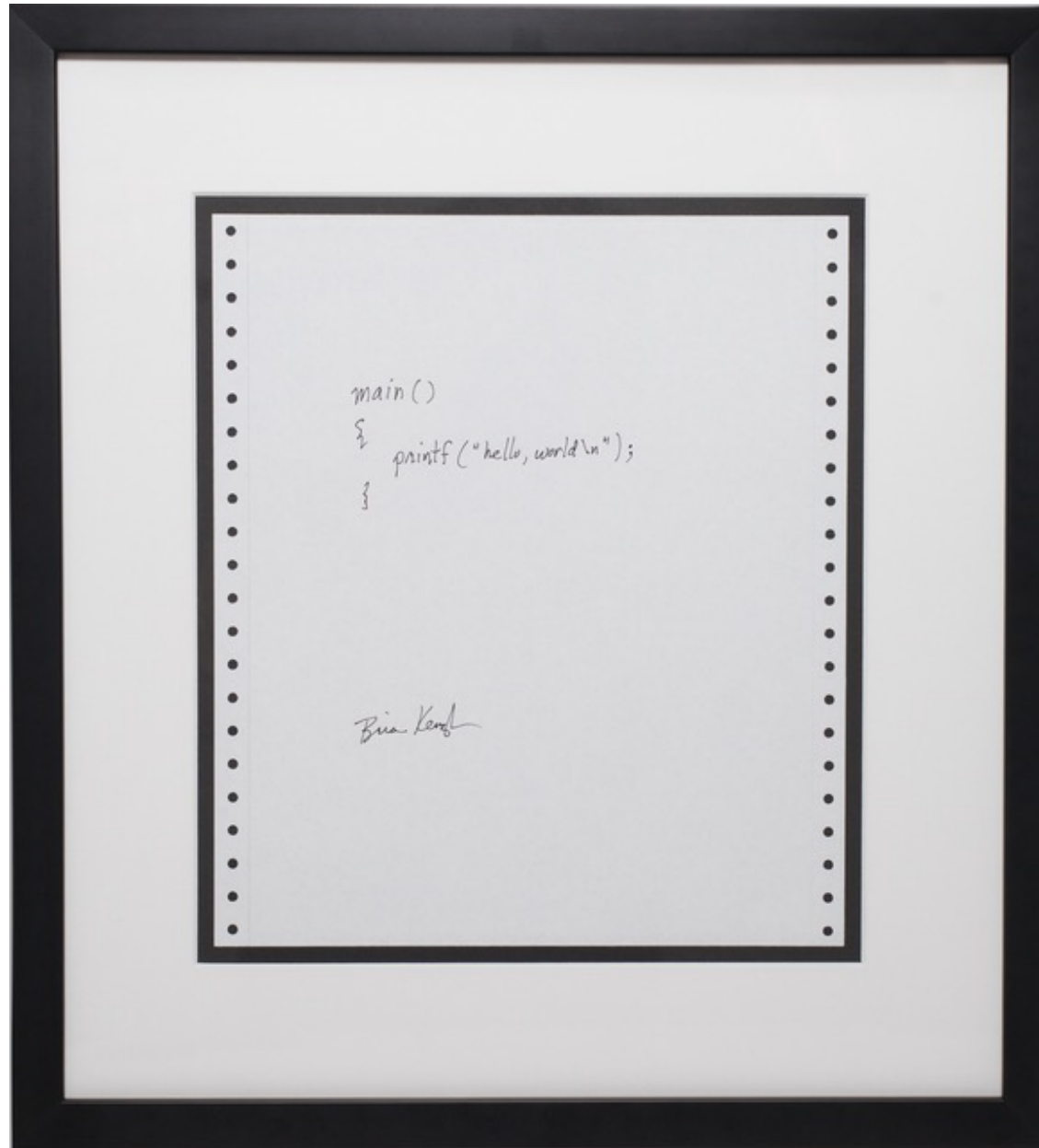
Publisher: Pearson

2nd edition (January 2004, reprint 2018)

Pages: 313

Euro: 31,45 (Amazon IT)

HELLO, WORLD



ALTRI LIBRI



ITA

C didattica e programmazione

By A. Kelly, I. Pohl

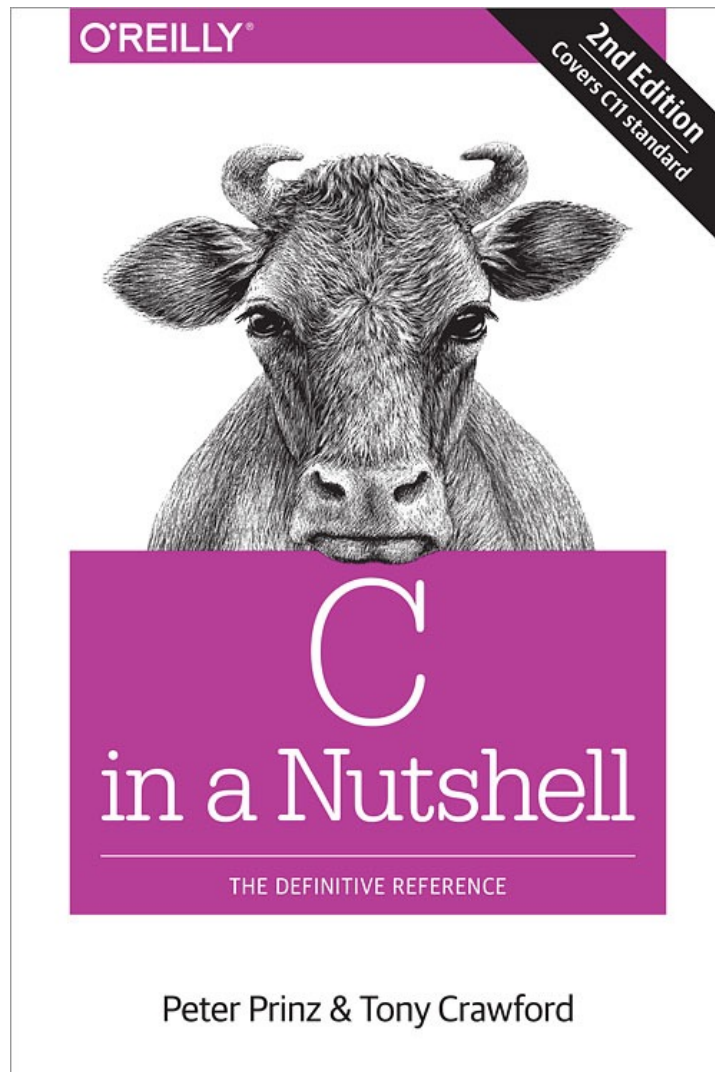
Publisher: Pearson

2nd edition

Pages: ~672

Euro: ~41,80 (Amazon IT)

ALTRI LIBRI

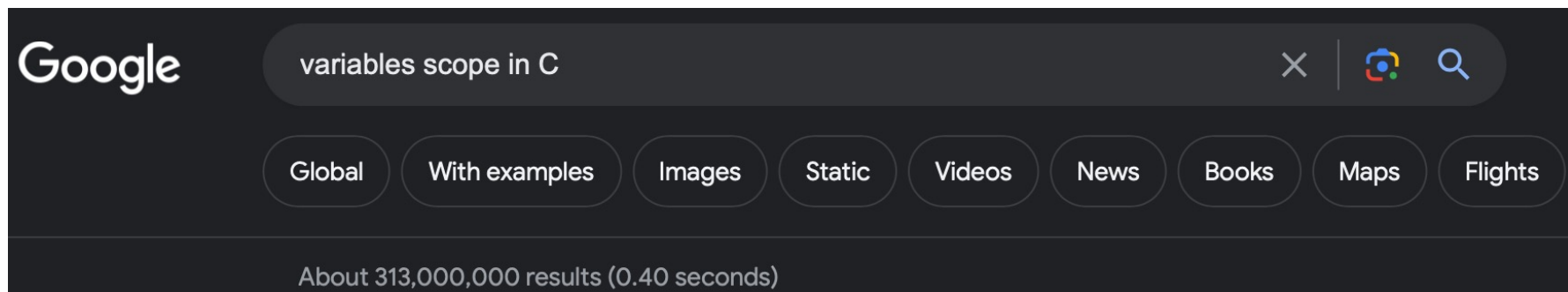
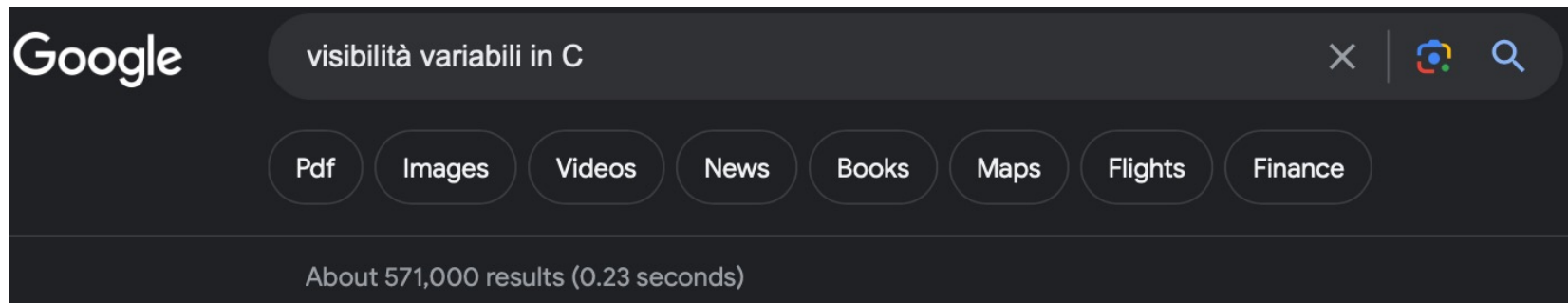


EN

C in a nutshell

By P. Prinz, T. Crawford
Publisher: O'Reilly
2nd edition (March 2016)
Pages: 812
Euro: 61,94 (Amazon IT)

WHY WILL THE SLIDES BE IN ENGLISH?



DOVE AIUTARSI

@ <https://stackoverflow.com>

c

C is a general-purpose programming language used for system programming (OS and embedded), libraries, games, and cross-platfor...

406523 questions 18 asked today, 136 this week

@ <http://www.cprogramming.com/tutorial/c-tutorial.html>

@ <http://www.w3schools.in/c-tutorial/>

@ <https://www.tutorialspoint.com/cprogramming/>

python

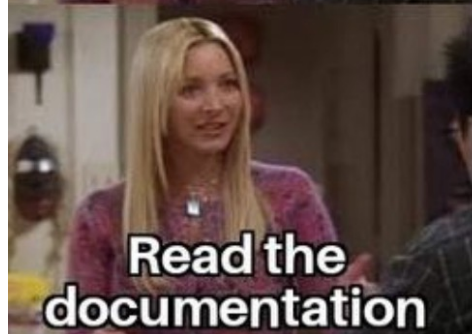
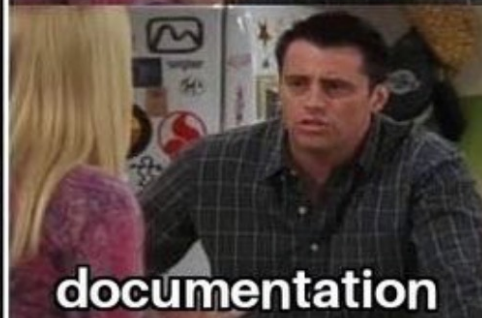
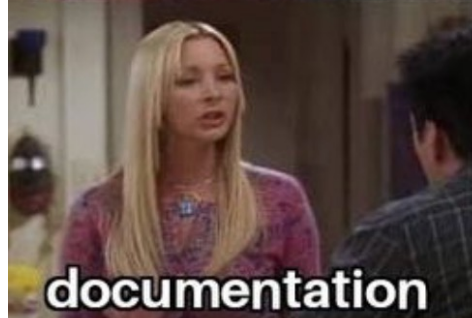
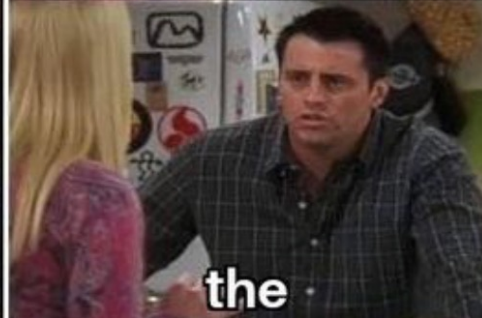
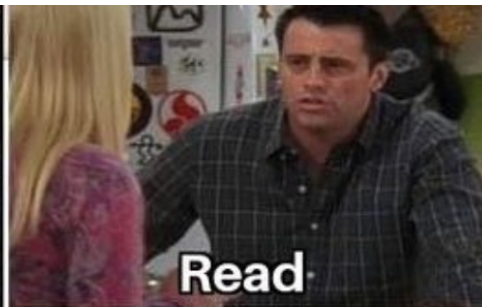
Python is a dynamically typed, multi-purpose programming language designed to be quick to learn, understand, and use, with a clean...

2210655 questions 155 asked today, 1415 this week

c#

C# (pronounced "see sharp") is a high-level, statically typed, multi-paradigm programming language developed by Microsoft. C# code...

1622570 questions 42 asked today, 510 this week



I heard he can code without Stack Overflow



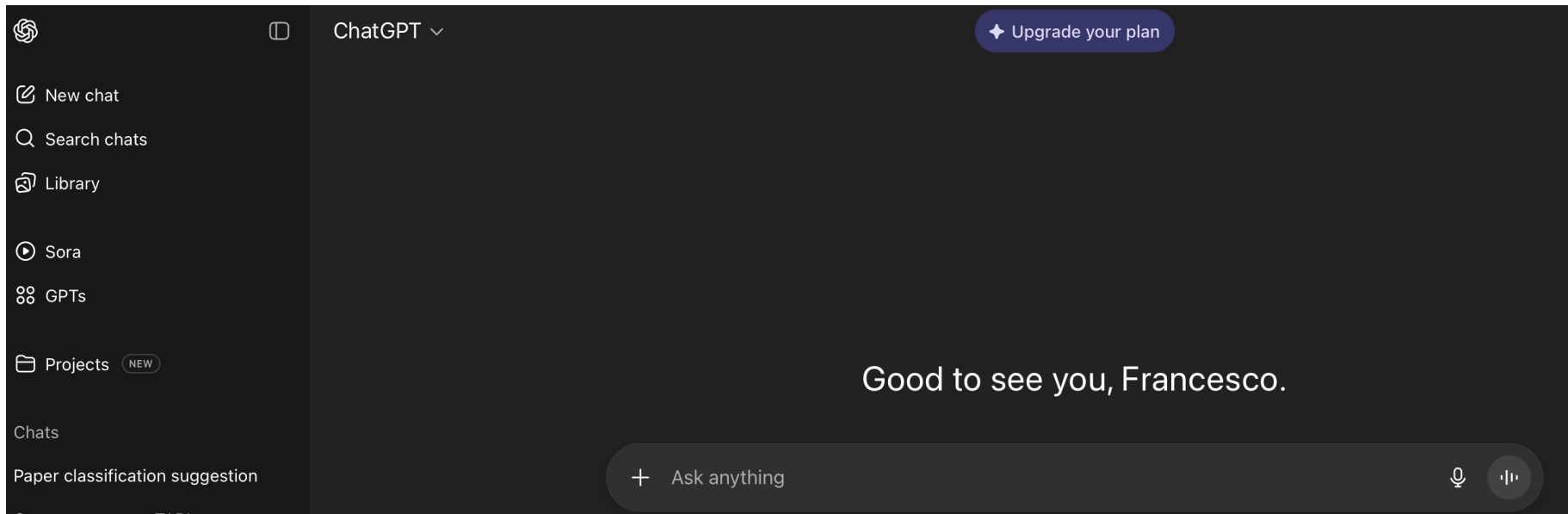
son : dad, why is my sister's name rose

dad : because your mom loves roses

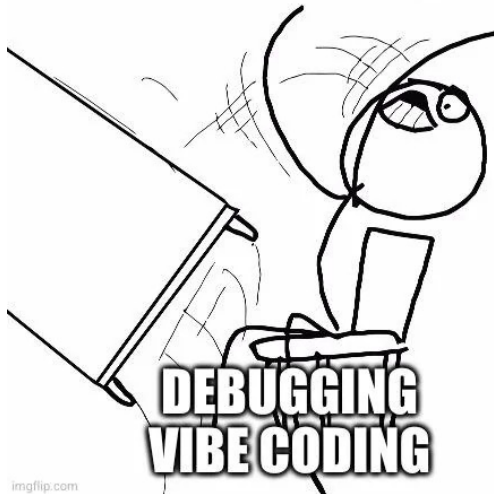
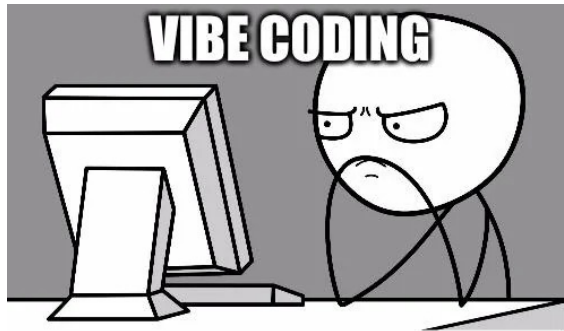
son : okay dad

dad : no problem, stack overflow





Gemini, Github Copilot, etc



Il **vibe coding** è un modo informale di programmare, basato su intuito e sperimentazione invece che su piani rigidi.

Si “va a sensazione”, spesso con l’aiuto di strumenti AI, lasciando che il codice prenda forma in modo creativo.

ESAME

1. Progetto in C da consegnare su [GitHub Classroom](#)
 - ✓ Assegnato a Dicembre
 - ✓ La scadenza per consegnare ciascun progetto è due giorni dopo circa la prova scritta (chiusura automatica su GitHub)
 2. Prova scritta
 3. Prova di Laboratorio su progetto consegnato + orale
- 📍 Registrazione su SOL: <https://unipg.esse3.cineca.it>
- 📍 Date:
- ✓ 12 Gennaio, 27 Gennaio, 10 Febbraio, ecc (3 estate, 2 Settembre: 8 appelli)

ESAMI PASSATI CON CORREZIONI

© <http://www.dmi.unipg.it/francesco.santini/progl.html>

ESEMPIO

Prova scritta Programmazione Procedurale - 14 Settembre 2022

Nome e Cognome: _____ Matricola: _____

1. **6 punti** Cosa stampa il programma? Gli operatori unari hanno precedenza massima, a si trova all'indirizzo di memoria `0x7fff54824ffa`, uno `short` occupa 2 byte, un `int` 4 byte, un `long` 8 byte. Suporre anche che, se l'operando sinistro di un operatore `||` è vero, l'operando destro NON viene valutato.

```
1 int a = 3, *b= &a;
2 int c = !(a-=1, ((a= 6) || ++a));
3 int d = !c && (a-=1, (a= 6) && ++a);
4 printf( "%d %d %d\n", a, c, d);
5 printf("%p %p %lu\n", b, (long *) (
    short *) b + 1, sizeof(*b));
```

2. **6 punti** Su foglio protocollo scrivere una funzione che crea un array di n elementi (con n intero senza segno passato come parametro) e lo inizializza con una successione di potenze di due. Per esempio se $n = 5$ l'array deve contenere 1-2-4-8-16-32. L'array creato deve essere poi ritornato dalla funzione come risultato.
3. **6 punti** Data la seguente `struct Node` definire su foglio protocollo una funzione di nome `cancella_2_posizione()` che cancella sempre il secondo elemento della lista (se presente) mantenendo la lista connessa: se la lista originale è 7-4-11, la nuova lista sarà 7-11. Supporre di avere un puntatore ad inizio lista globale di nome `pFirst`.

```
1 struct Node {
2     int info= 0;
3     struct Node* pNext= NULL;
4 }
```

4. **5 punti** Elencare quali dei seguenti comandi provocano errore a causa del linker (e perché): 1) `gcc -o write write.c`, 2) `gcc -c main.c`, 3) `gcc -o main main.c`, 4) `gcc write.c main.c -o main`. In caso il punto 4) ritorni un errore, descrivere come può essere corretto. Infine, che tipo di `linkage` hanno `count`, `i`, e `mywrite`? Cosa stampa il programma `main`?

```
main.c
int count;
int count= 3;
void mywrite(int count);
int main(void){
extern int count;
do {
    mywrite(count);
}while(count >=0);
}

write.c
#include <stdio.h>
int i = 1;
static int count= 4;
void mywrite(int a){
    a--;
    printf("%d\n", count= count - i);
}
```

5. **7 punti** Cerchiare le affermazioni vere dato `int a[5]= {129, INT_MIN, INT_MIN | INT_MAX, 262142, 262168}; q[1]= 1; short int *p = (short*) a; char *q= (char*) a;` sapendo che i tre tipi usati occupano 4, 2, e 1 byte, e $262144 = 2^{18}$ (valori rappresentati in *little endian* e complemento a due). Scrivere la mappa di memoria e giustificare le affermazioni (vere o false). A. $(*(p + 5) - p[6]) \% 1$ B. $\&a[5] - (a + 2) - q[1] - 2$ C. $*((short*)(q + 13)) == *((short*)(q + 17))$

REGOLE

- Ⓢ Nome e cognome su testo e tutti i fogli protocollo usati
- Ⓢ Scrivere svolgimento su foglio protocollo
 - ✓ Senza un'idea di svolgimento vale 0 punti
- Ⓢ Copiare solo la soluzione nel riquadro dell'esercizio corrispondente nel testo
 - ✓ Oppure "Vedi foglio"
- Ⓢ Durata ~ 1 or e 45 minuti

- Ⓢ Sul banco, consentito solo materiale per scrivere

- Ⓢ Ammessa calcolatrice NON scientifica

PROGETTO

- Ⓢ Progetto INDIVIDUALE valutato da 1 a 4 (+ se aggiunte altre feature), ma
 - ✓ Se non compila a causa di errori, valutato non sufficiente: no ammessi all'orale
 - ✓ Se alcune funzioni sono sbagliate, valutato non sufficiente: no ammessi all'orale
- Ⓢ Punti sommati al voto dello scritto
- Ⓢ In generale il progetto riguarda lo sviluppo di varie funzioni su una lista dinamica, ma non sempre (vedi anno scorso)
- Ⓢ Prova anti-plagio

ANTIPLAGIO

ESAME PROGRAMMAZIONE I

[[How to Read the Results](#) | [Tips](#) | [FAQ](#) | [Contact](#) | [Submission Scripts](#) | [Credits](#)]

File 1	File 2	Lines Matched
C:/ProgrammazioneI-UniPG/prova-finale-di-programmazione-secondo-appello/██████c98/ (96%)	C:/ProgrammazioneI-UniPG/prova-finale-di-programmazione-secondo-appello/██████82/ (97%)	1035
C:/ProgrammazioneI-UniPG/prova-finale-di-programmazione-secondo-appello/██████ni4/ (74%)	C:/ProgrammazioneI-UniPG/prova-finale-di-programmazione-secondo-appello/██████elli/ (78%)	776
C:/ProgrammazioneI-UniPG/prova-finale-di-programmazione-secondo-appello/██████ce/ (96%)	C:/ProgrammazioneI-UniPG/prova-finale-di-programmazione-secondo-appello/██████ardo/ (97%)	669
C:/ProgrammazioneI-UniPG/prova-finale-di-programmazione-secondo-appello/██████fa/ (85%)	C:/ProgrammazioneI-UniPG/prova-finale-di-programmazione-secondo-appello/██████p0/ (92%)	831
C:/ProgrammazioneI-UniPG/prova-finale-di-programmazione-secondo-appello/██████etta/ (84%)	C:/ProgrammazioneI-UniPG/prova-finale-di-programmazione-secondo-appello/██████etti/ (86%)	759
C:/ProgrammazioneI-UniPG/prova-finale-di-programmazione-secondo-appello/██████etto/ (76%)	C:/ProgrammazioneI-UniPG/prova-finale-di-programmazione-secondo-appello/██████sae/ (80%)	722
C:/ProgrammazioneI-UniPG/prova-finale-di-programmazione-secondo-appello/██████c0/ (80%)	C:/ProgrammazioneI-UniPG/prova-finale-di-programmazione-secondo-appello/██████sae/ (78%)	729

Niente orale per entrambi

PROVA LABORATORIO

1. Verra richiesto di modificare il progetto assegnato a casa

🌀 Esempio di modifica

✓ Creare una nuova funzione ed aggiungerla al menu in modo da [funzione su lista dinamica]

🌀 Durata: 1 ora e mezzo

2. Alcune domande sul programma svolto

✓ Max due punti in più rispetto al voto dello scritto (ma anche n punti in meno se non si sa rispondere)

**SE NON IN GRADO MODIFICARE IL
VOSTRO PROGETTO NEL TEMPO
ASSEGNATO, ESAME FALLITO**



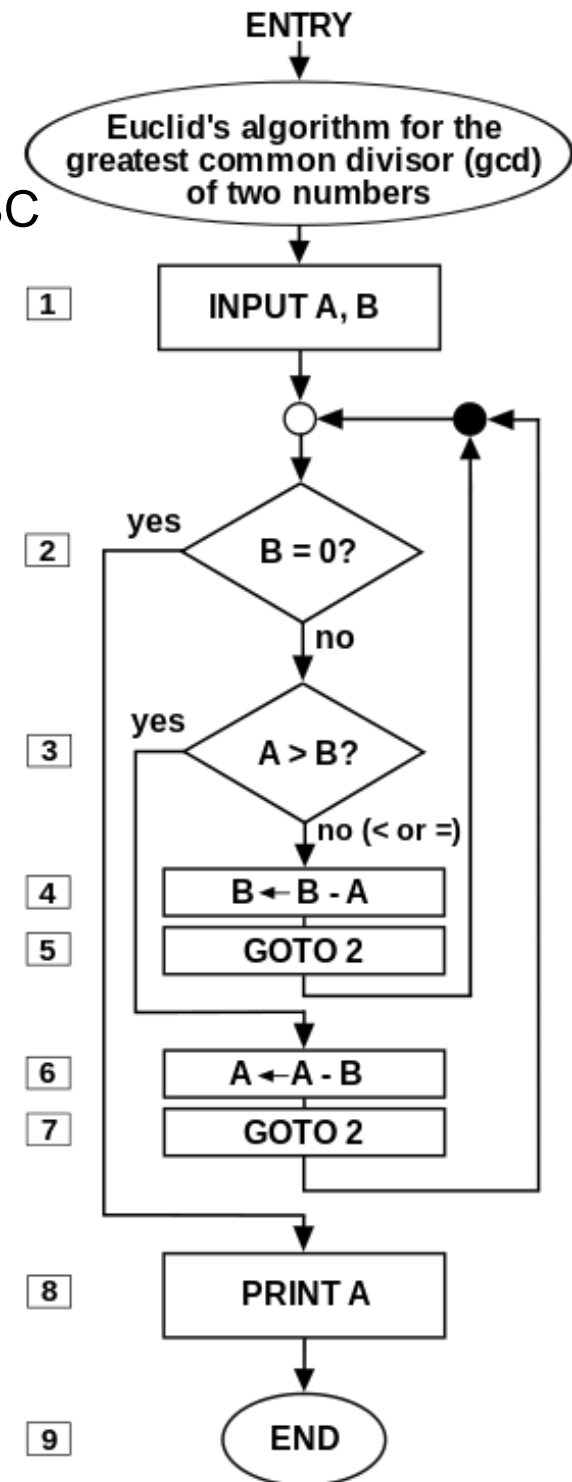
LET'S START



ALGORITHMS

Euclid's *Elements* 300 BC




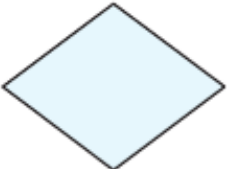

- ① A procedure for solving a mathematical problem (as of finding the greatest common divisor) in a finite number of steps that frequently involves repetition of an operation.
- ② A **flowchart** is a type of diagram that represents an algorithm, showing the steps as boxes of various kinds, and their order by connecting them with arrows.
- ③ Sequence of operations from top to bottom



WHERE ALGORITHM COMES FROM

- ④ It comes from Al-Khwārizmī (Persian: خوارزمی, c. 780–850), a Persian mathematician, astronomer, geographer, and scholar.

ANSI/ISO 1970 (REVISED 1985)

Symbol	Name	Description
	Terminator	Represents the start or end of a program or module
	Process	Represents any kind of processing function; for example, a computation
	Input/output	Represents an input or output operation
	Decision	Represents a program branch point
	Connector	Indicates an entry to, or exit from, a program segment

EXAMPLE OF A PROGRAM IN C

```
#include <stdio.h>

int main()
{
    int a, b;

    printf("Enter first positive integer: \n");
    scanf("%d", &a);
    printf("Enter second 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;
}
```

<https://www.youtube.com/watch?v=EbG9RjFlr4g>

THE FRIENDSHIP ALGORITHM

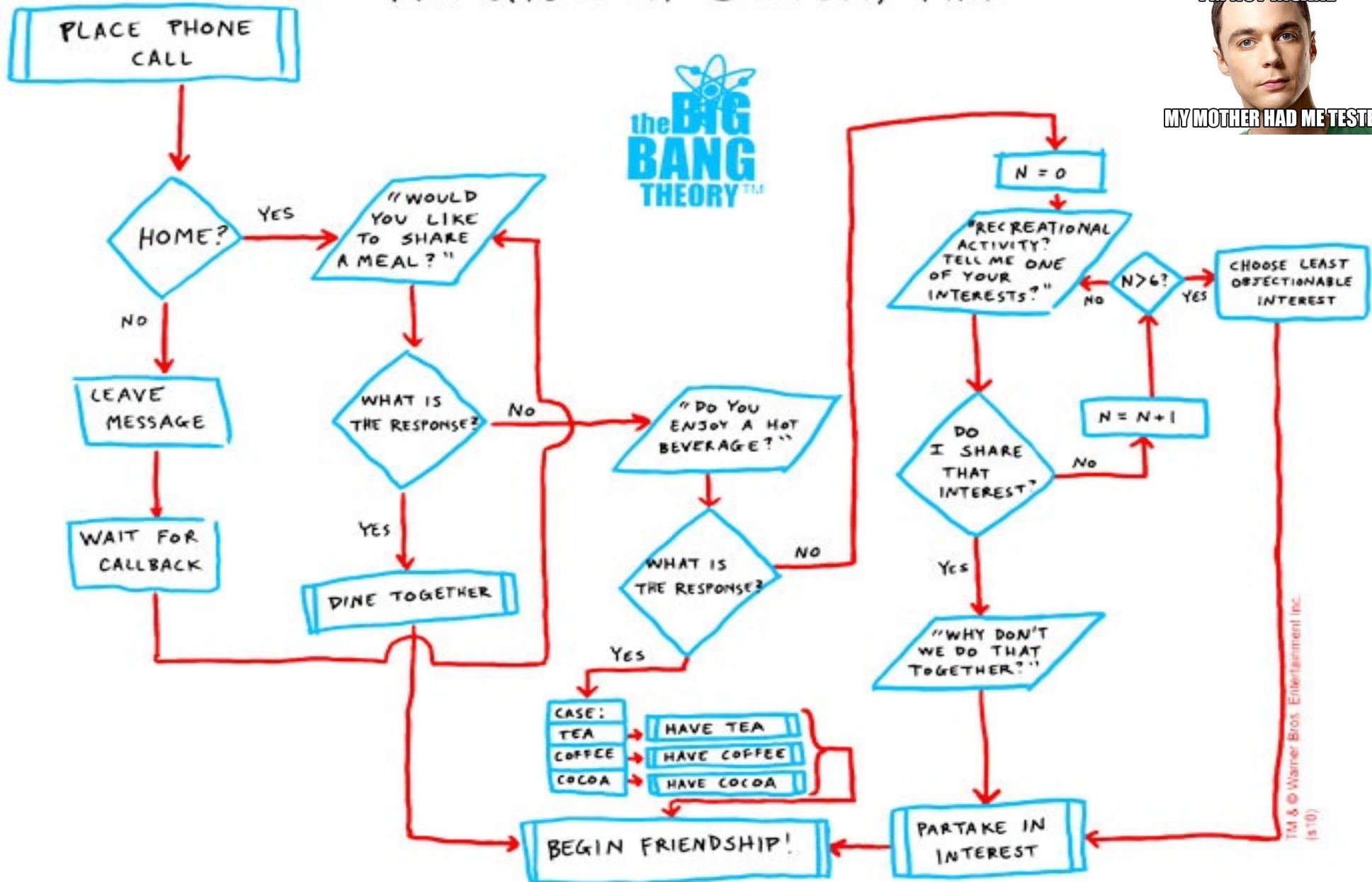
DR. SHELDON COOPER, Ph.D



I'M NOT INSANE



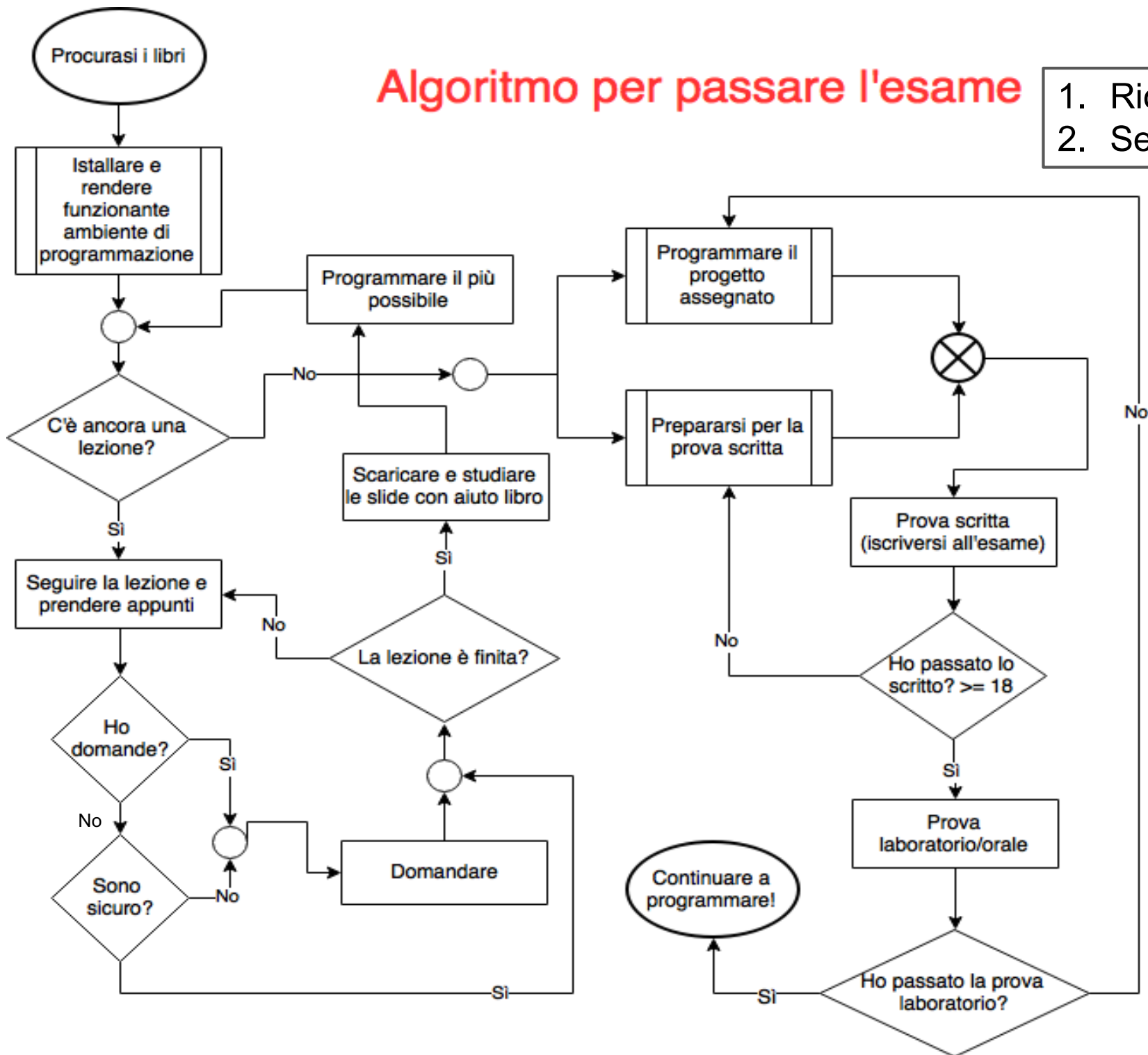
MY MOTHER HAD ME TESTED



TM & © Warner Bros. Entertainment Inc. (s10)

Algoritmo per passare l'esame

1. Ricevimento
2. Seconda prova



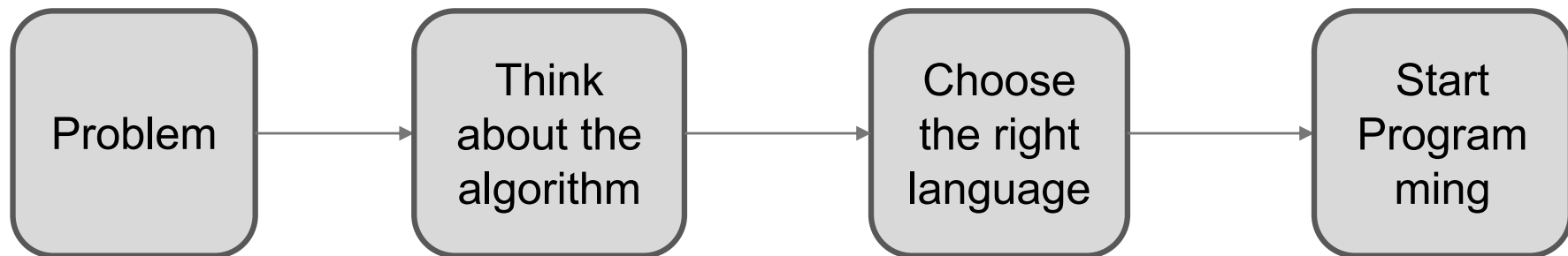
FLOWCHART RESOURCES

@ <https://en.wikipedia.org/wiki/Flowchart>

@ <https://www.draw.io>

PROGRAMMING

- ④ **Programming** is the process of taking an algorithm and encoding it into a notation, a programming language, so that it can be executed by a computer.
- ④ Problem solving
- ④ A program is a collection of instructions



TIPS: HOW TO LEARN PROGRAMMING

- 1. Look at the Example Code:** When you're first learning to program, you should make sure to look at, and try to understand, every example. Read the code examples before the text, and try to figure out what they did. (NOW)
- 2. Don't Just Read Example Code--Run It:** Then **type the sample code into a compiler**--if you type it, instead of copying and pasting it, you will really force yourself to go through everything that is there. (NOW)
- 3. Write your Own Code as Soon as Possible:** start writing sample programs that use every point we did. (NOW)
- 4. Seek out More Sources:** tutorials, examples, books, man. Program code ALWAYS with handbook and/or Internet. (NOW)

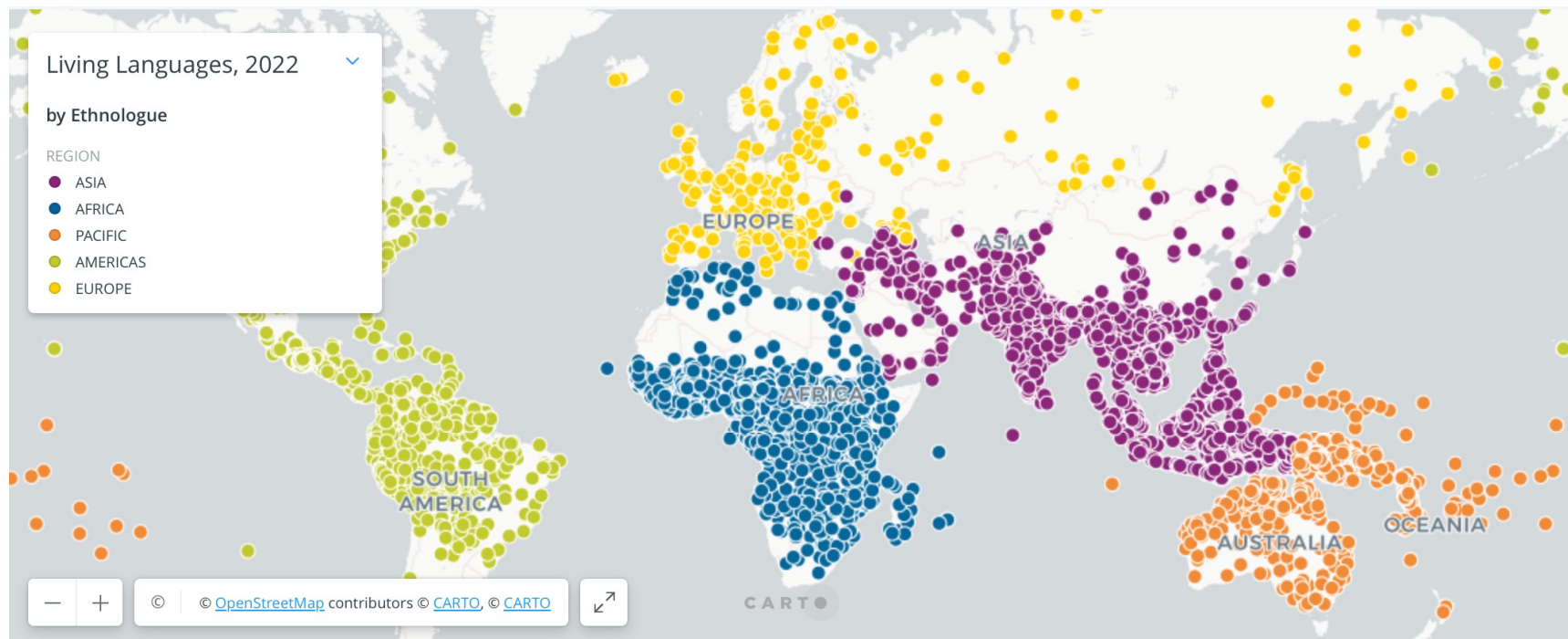
WHY C



HOW MANY SPOKEN LANGUAGES

@ ~7159

@ <https://www.ethnologue.com/guides/how-many-languages>



HOW MANY PROGRAMMING LANGUAGES?

🌐 Wikipedia has a list of 700 programming languages

✓ https://en.wikipedia.org/wiki/List_of_programming_languages

🌐 Tiobe has a list of over 265 languages

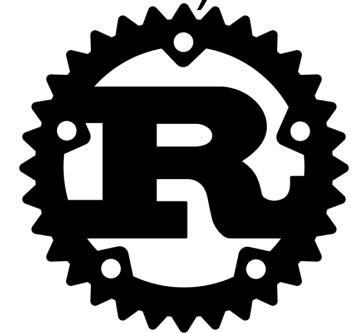
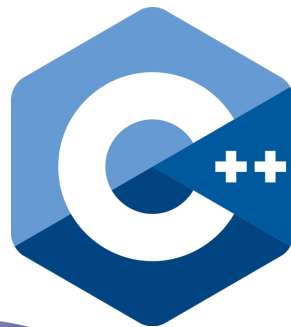
✓ TIOBE tracks a programming language if it passes 3 tests: it must have its own Wikipedia page, it must be Turing complete, and a Google search for it must return over 5,000 search results.

✓ https://www.tiobe.com/tiobe-index/programminglanguages_definition/











WHAT IS THE BEST LANGUAGE?

@ It depends:

- ✓ Web: **Javascript**
- ✓ (data) science/mining, machine learning: **Python**
- ✓ Low level and fast applications: **C/C++**
- ✓ Write once run everywhere, Android: **Java**
- ✓ Server-side for the Web: **PHP**
- ✓ Productivity for backend development (Airbnb, GitHub):
Ruby



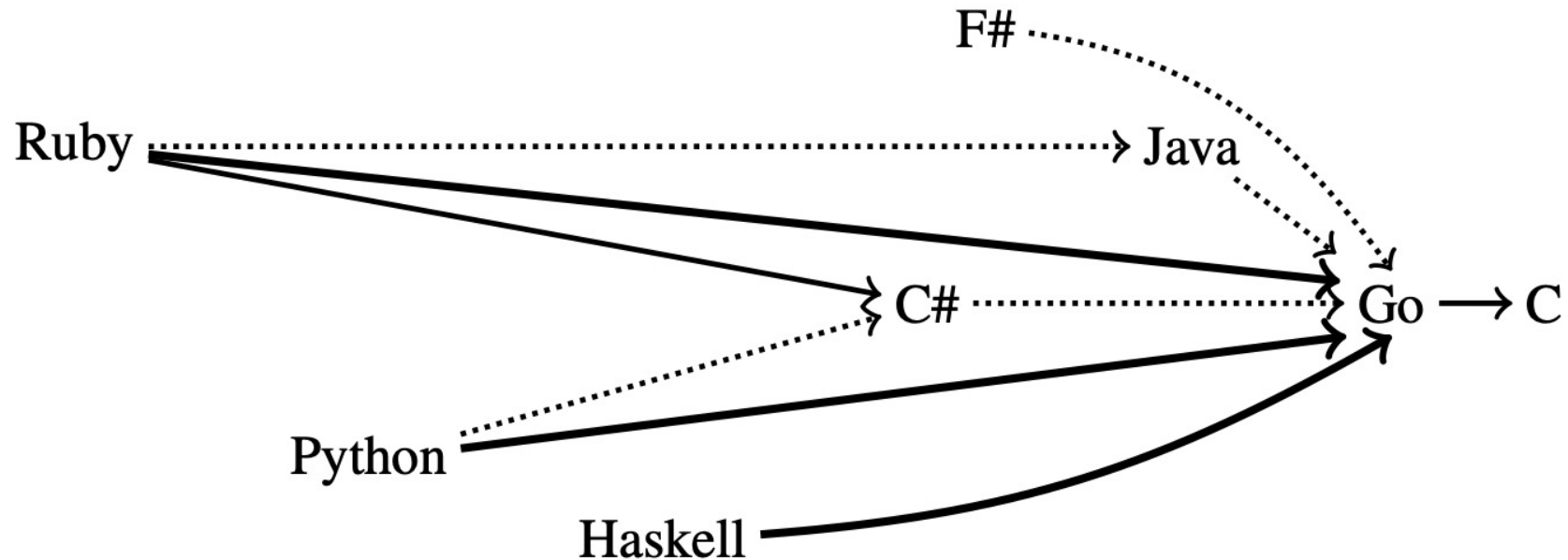
TIOBE INDEX (POPULARITY)

Sep 2025	Sep 2024	Change	Programming Language	Ratings	Change
1	1		 Python	25.98%	+5.81%
2	2		 C++	8.80%	-1.94%
3	4	▲	 C	8.65%	-0.24%
4	3	▼	 Java	8.35%	-1.09%
5	5		 C#	6.38%	+0.30%
6	6		 JavaScript	3.22%	-0.70%
7	7		 Visual Basic	2.84%	+0.14%
8	8		 Go	2.32%	-0.03%
9	11	▲	 Delphi/Object Pascal	2.26%	+0.49%
10	27	▲▲	 Perl	2.03%	+1.33%

The ratings are based on the number of skilled engineers world-wide, courses and third party vendors. Popular search engines such as Google, Bing, Yahoo!, Wikipedia, Amazon, YouTube and Baidu are used to calculate the ratings.

<https://www.tiobe.com/tiobe-index/>

C PERFORMANCE



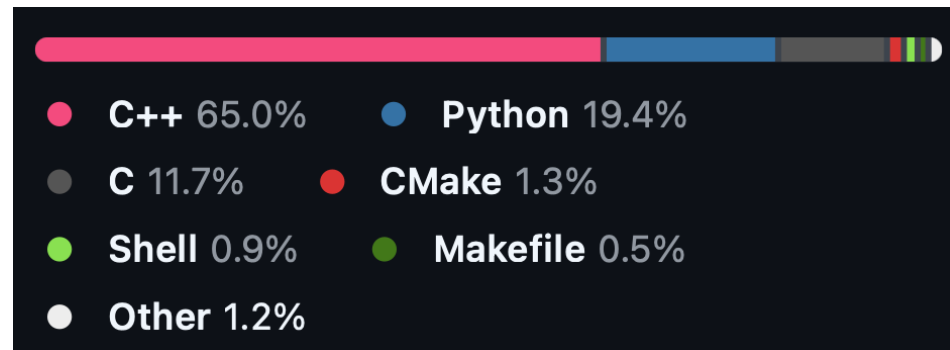
https://web.archive.org/web/20200709215859id_/https://bugcounting.net/pubs/icse15.pdf

<https://www.geeksforgeeks.org/blogs/top-10-fastest-programming-languages/>

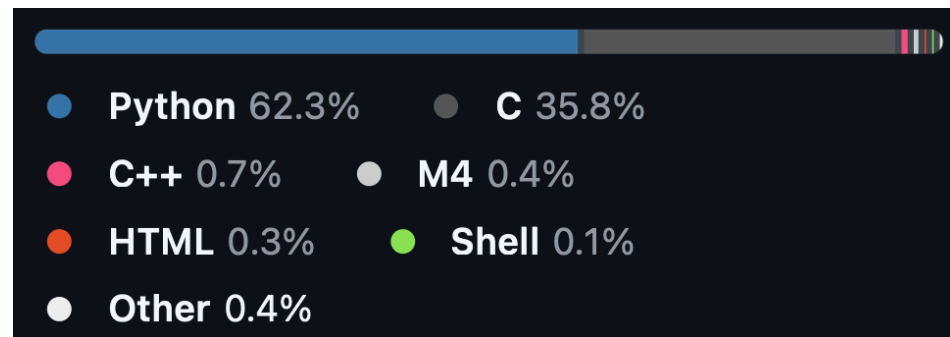
ENGINES

@ <https://bitcoincore.org>

@ <https://github.com/bitcoin/bitcoin>

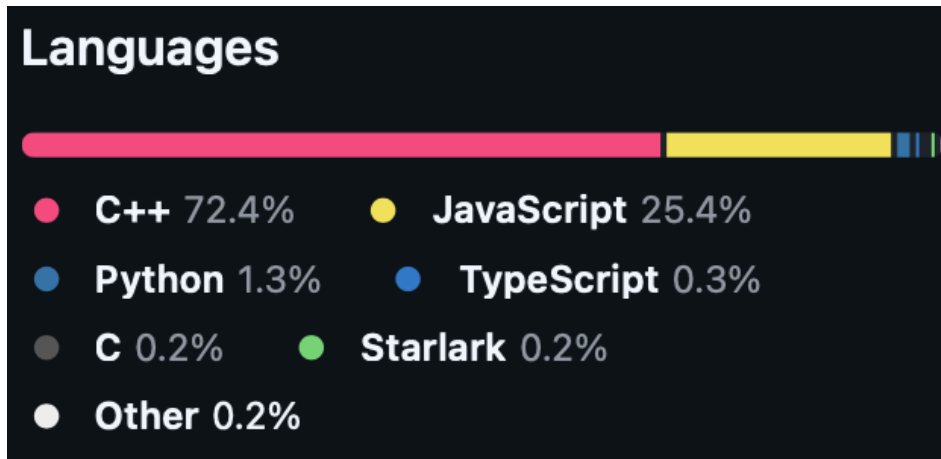


@ <https://github.com/python/cpython>



ENGINES


@ <https://github.com/v8/v8>



JOBS



www.glassdoor.com/job-listing/ai

LAIKA Search Jobs - Google Careers FCA Fiat Chrysler Automobiles Automated Driving Software Engineer...




Automated Driving Software Engineer

3.4 ★ FCA Fiat Chrysler Automobiles – Auburn Hills, MI


Apply Now  Save 

22 days ago

Company Rating

3.4 

Glassdoor Estimated Salary

\$87,000/year 

Job Company Reviews

The role will function as a member of the Automated Driving Component Software team. The candidate will be responsible for the development and integration of SAE Level 3/4 automated driving feature and functional software. The primary duties will include the development of infrastructure and basic software compliant with functional safety principals, development of middleware and software service modules and the development of a modular serial data build/parse architecture. The ideal candidate will possess a good understanding of modern safety critical electronic control units and possess an interest in state-of-the-art automated driving compute platforms and emerging computational technologies.

Basic Qualifications:

- Bachelor of Science degree in Applied Physics, Mechanical, Electrical, Aerospace, or Computer Engineering, Computer Science or related Engineering field
- Minimum of 3 years embedded or control systems software experience and embedded C/C++ programming
- Experience with CAN stack, SPI, LIN and Flash Bootloader integration, OSEK OS, AutoSAR Classic OS or diagnostic management
- Good communication skills and the ability to interface with HIL testing, vehicle integration and validation engineering organizations
- Familiarity with SW integration tools (e.g. Emulators, Debuggers, CANoe/CANalyzer)
- Familiarity with UDS, XCP and Instrumented Embedded Control Units
- Familiarity using trace capture tools
- Ability to interface with Tier I and Tier II automotive suppliers

Preferred Qualifications:

JOBS

The screenshot shows a web browser window with the URL `graphics.pixar.com/research`. The browser tabs include "LAIKA", "Search Jobs - Google Careers", and "Pixar Research Group". The website header features the Pixar Research Group logo with the letters "P", "A", and "R" and the text "RESEARCH GROUP". Below the header is a navigation menu with links for "HOME", "ABOUT Us", "PEOPLE", "JOBS", and "ONLINE LIBRARY".

Research Jobs

How to Apply

To apply for any of the positions listed here, visit the [Pixar Careers Page](#) then follow the instructions you'll find there.

Research Scientist

We are currently looking for [Research Scientists](#) at all levels. If interested, please submit an application at our [careers site](#).

Internships

We are always looking for interns to work closely with our research staff. We are flexible on start dates and are not limited to the summer quarter. Many internships are one quarter in duration, but we prefer longer stays when possible.

Who Should Apply

We seek exceptionally talented graduate students and advanced undergraduates to assist in the creation, implementation and transfer of new technology related to computer graphics film making. The position requires a deep understanding of mathematical algorithms, an ability to collaborate in small to medium sized groups consisting of world-class computer graphics researchers, and skill in rapidly implementing and testing new algorithms.

Qualifications

- Excellence in problem solving
- Strong math and computer graphics experience
- Strong communication skills.
- Strong software engineering skills with expertise in C and C++.
- Current standing as a graduate student or upper division undergraduate
- 1+ years of research experience
- Ability to work well with a wide range of personality types

JOBS

https://pixar.wd501.myworkdayjobs.com/en-US/Pixar_External_Career_Site/details/Software-Engineer-Tools-Internals--Core_R-03787-1

The screenshot shows a web browser window with the following elements:

- Address bar: `https://pixar.wd501.myworkdayjobs.com/en-US/Pixar_External_Career_Site/details/Software-Engineer-Tools-Internals--Core_R-03787-1`
- Page title: `Ubisoft - Careers / Search for Jobs`
- Search bar: `13 matches` with a search input containing `programm` and a `Done` button.
- Navigation menu: `UBISOFT`, `GAMES`, `STORE`, `COMMUNITY`, `COMPANY`, `CAREERS`, `English`, and `SEARCH`.
- Buttons: `BACK TO SEARCH`, `I'M INTERESTED`, and `REFER A FRIEND`.
- Section header: `Lead Programmer (156) (Programming)`
- Location and details: `Newcastle upon Tyne, United Kingdom - Full-time - REF7601C`
- Section header: `JOB DESCRIPTION`
- Text: `About Reflections`, `Reflections, a Ubisoft studio is a video games development company based in Newcastle, UK. Collaborating with Ubisoft's international creative teams on projects such as Assassins Creed Syndicate, Tom Clancy's Ghost Recon, online open-world RPG "The Division" as well as the adventure platformer "Grow Home". With a host of other top titles to our name such as open-world action adventure "Watch Dogs", MMO racer "The Crew" and the latest instalment in the world's No.1 dance game franchise 'Just Dance 2015' Reflections' objective is to combine technical expertise, flair and innovation to create and contribute to successful and memorable games. We are looking for Pioneering, Obsessive Humans to cement its reputation as an industry leading studio within the UK and the world.`
- Text: `Job Purpose`, `Be responsible for leading the engineering team within a collaborative multidiscipline environment, defining and tracking the schedule for his team, managing and mentoring individuals within the engineering team, and being an evangelist for the project.`
- Text: `Reports To`, `Producer of the Project`
- Section header: `QUALIFICATIONS`
- Section header: `Skills and Knowledge`
- List-Group:
 - Passionate about playing and making games, and has a good knowledge of the games industry in general;
 - Excellent C/C++ programming skills, with excellent knowledge of object oriented development;
 - Familiarity with common scheduling, task and issue tracking tools – MS Project, Jira, etc;
 - Able to create, manage and track schedules for a team of engineers;

Skills and Knowledge

- Passionate about playing and making games, and has a good knowledge of the games industry in general;
- Excellent C/C++ programming skills, with excellent knowledge of object oriented development;
- Familiarity with common scheduling, task and issue tracking tools – MS Project, Jira, etc;
- Able to create, manage and track schedules for a team of engineers;

OTHER SUGGESTED LANGUAGES

@ Python

@ Javascript (Java)

SU LIBRO

📍 Sezioni 1.1-1.5

📍 Sezione 1.8

