INFO

(Programmazione Procedurale con Laboratorio: 73 ore (9 CFU, 7+2)

Pagina Web corso: http://www.dmi.unipg.it/francesco.santini/progI.html

Pagina Unistudium: https://www.unistudium.unipg.it

Orario Martedì 14-16 (I1), Mercoledì 9-11 (I1), Giovedì 11-13 (A2), Venerdì 11-13 (A2)

Canali per comunicare (più lenti):

✓ Ricevimento su appuntamento via email o telegram
✓ francesco.santini@unipg.it
TELEGRAM (PIÙ VELOCE)

Telegram: @safran
Canale: https://telegram.me/programmazione1_PG

Usato come metodo veloce per comunicare con me (@safran) e per comunicazioni generali (il canale)

ISCRIVETEV I !!!
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
PROGRAMMA

- General programming concepts
  - Interpreted and compiled languages
  - Scope
  - Recursion(?)
- Programming in C part II
  - Compiling with GCC
  - Debugging C Programs with GDB
- 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

- Slide (su mia pagina Web e Unistudium), upload PRIMA della lezione
- Libri (proposti successivamente)
- Esercitazioni
Il Linguaggio C – Fondamenti e tecniche di programmazione
By Paul J. Deitel and Harvey M. Deitel
Publisher: Pearson, 8th edition
(August 2016)
Pages: 637
Euro: 33,15 (Amazon)
C in a Nutshell (Pocket Reference)
By Peter Prinz, Ulla Kirch-Prinz
Publisher: O'Reilly Media
Final Release Date: November 2002
Pages: 144
Euro: 8,32 (Amazon)
Il linguaggio C. Principi di programmazione e manuale di riferimento

By B. Kernighan, D. Ritchie
Publisher: Pearson
2nd edition (January 2004)
Pages: 313
Euro: 22,95 (Amazon IT)
HELLO, WORLD

```c
#include <stdio.h>

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

Ewan Kerr
C didattica e programmazione

By A. Kelly, I. Pohl
Publisher: Pearson
2nd edition
Pages: ~672
Euro: ~37 (Amazon IT)
C in a nutshell

By P. Prinz, T. Crawford
Publisher: O'Reilly
2nd edition (March 2016)
Pages: 812
Euro: 54 (Amazon IT)
WHY WILL THE SLIDES BE IN ENGLISH?

visibilità variabili in C

About 1,270,000 results (0.34 seconds)

scope variable in C

About 157,000,000 results (0.51 seconds)
DOVE AIUTARSI

https://stackoverflow.com

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

http://www.cprogramming.com/tutorial/c-tutorial.html

http://www.w3schools.in/c-tutorial/

https://www.tutorialspoint.com/cprogramming/
ESAME

1. Progetto in C da consegnare su GitHub Classroom
   ✓ Assegnato a Dicembre
   ✓ La scadenza per consegnare ciascun progetto è qualche giorno dopo 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/Home.do

Date:
✓ 12 Gennaio, 26 Gennaio, 9 Febbraio, 13 Aprile, 15 Giugno, 6 Luglio, 14 Settembre, 9 Novembre
ESAMI PASSATI CON CORREZIONI

http://www.dmi.unipg.it/francesco.santini/progl.html
ESEMPIO

Prova scritta Programmazione I - 5 Luglio

Nome e Cognome: 
Matricola: 

1. **2 punti** Elencare tutte le conversioni di tipo. Quanto vale a alla fine?
   1. #define A 2.5
   2. short int f(double p1, short int p2)
      3. return (p1 => p2 ? p1 : p2);
   4. }
   5. int main(void) {
      6. float a = A;
      7. int b = 2LL;
      8. a = f(a, b);
      9. }

2. **3 punti** Scrivere cosa stampa il seguente programma.
   1. int i, a;
   2. for (i = 0; i < c; i++)
      3. if (i == 0 || i == c - 1) {
         4. printf("%d \n", a, b);
         5. }
   6. }
   7. printf("%d \n", a, b);

3. **4 punti** Scrivere cosa stampa il seguente programma.
   1. int c = 5, c = 7, 1, 1;
   2. for (i = 0; i < c; i++)
      3. if (i = 0 || i = c - 1) {
         4. printf("0");
         5. continue;
         6. }
   7. else
   8. printf("1");
   9. }
   10. printf("\n");

4. **4 punti** Cerchiera tra i seguenti punti, quelli che rappresentano un realue, dato int a[5]; int *p = a; A. a[1] B. a[0]+1 C. p D. *p E. p+1 F. p+1 G. 7 H. (&a[4]) a = a[1]

5. **3 punti** Cosa stampa il programma? a si trova all’indirizzo di memoria 0x7f7f5eb9f6f5, un long long occupa 8 byte, un int 4 byte.
   1. long long int a = -3, *b = &a;
   2. int c = (a==1+1, a + 1, a ? (a + 1), 1) ;
   3. printf("%lld %lld %lld\n", a, c);
   4. printf("%lp %lp %lp\n", b, (int *) b + 1, 
   5. sizeof(*b));
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 ~ 2:20 ore

 Sul banco, consentito solo materiale per scrivere.
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
Niente orale per entrambi
1. Verrà richiesto di modificare il progetto assegnato a casa

- Esempio di domande
  - Creare una nuova funzione che invece di inserire in testa alla lista (come nel testo del progetto) inserisce in fondo alla lista
  - Creare una funzione che scorre la lista e aggrega dei campi: per esempio somma il campo X per tutti gli elementi della lista

- Durata: 1 ora

2. Alcune domande

  - Alcuni punti in più (o in meno)
SE NON IN GRADO MODIFICARE IL VOSTRO PROGETTO NEL TEMPO ASSEGNATO, ESAME FALLITO
LET’S START
ALGORITHMS

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.

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.

Sequence of operations from top to bottom

Euclid’s *Elements* 300 BC
WHERE ALGORITHM COMES FROM

It comes from Al-Khwārizmī (Persian: خوارزمی, c. 780–850), a Persian mathematician, astronomer, geographer, and scholar.
# ANSI/ISO 1970 (REVISED 1985)

<table>
<thead>
<tr>
<th>Symbol</th>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><img src="image1.png" alt="Terminator" /></td>
<td>Terminator</td>
<td>Represents the start or end of a program or module</td>
</tr>
<tr>
<td><img src="image2.png" alt="Process" /></td>
<td>Process</td>
<td>Represents any kind of processing function; for example, a computation</td>
</tr>
<tr>
<td><img src="image3.png" alt="Input/output" /></td>
<td>Input/output</td>
<td>Represents an input or output operation</td>
</tr>
<tr>
<td><img src="image4.png" alt="Decision" /></td>
<td>Decision</td>
<td>Represents a program branch point</td>
</tr>
<tr>
<td><img src="image5.png" alt="Connector" /></td>
<td>Connector</td>
<td>Indicates an entry to, or exit from, a program segment</td>
</tr>
</tbody>
</table>
#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;
}
THE FRIENDSHIP ALGORITHM

DR. SHELDON COOPER, PH.D

PLACE PHONE CALL

HOME? YES

"WOULD YOU LIKE TO SHARE A MEAL?"

WHAT IS THE RESPONSE? NO

DO YOU ENJOY A HOT BEVERAGE?"

WHAT IS THE RESPONSE? NO

WHY DON'T WE DO THAT TOGETHER?"

BEGIN FRIENDSHIP!

CASE: TEA, COFFEE, COCOA

HAVE TEA, COFFEE, COCOA

PARTAKE IN INTEREST

RECREATIONAL ACTIVITY? TELL ME ONE OF YOUR INTERESTS?"

N = 0

N > 6? NO

DO I SHARE THAT INTEREST? NO

N = N + 1

CHOOSE LEAST OBJECTIONABLE INTEREST

IM NOT INSANE

MY MOTHER HAD ME TESTED
Algoritmo per passare l'esame

1. Ricevimento
2. Seconda prova

1. Procurasi i libri
   - Istallare e rendere funzionante ambiente di programmazione

2. Programmare il progetto assegnato
   - Prepararsi per la prova scritta
     - Prova scritta (iscriversi all'esame)
       - Ho passato lo scritto? >= 18
         - Si
           - Prova laboratorio/orale
             - Ho passato la prova laboratorio?
               - No
                 - Continuare a programmare!
               - Si
                 - Ho passato la prova laboratorio?
                   - No
                     - Domandare
                       - Sono sicuro?
                         - No
                           - Si
                             - La lezione è finita?
                               - No
                                 - Scaricare e studiare le slide con aiuto libro
                                 - Seguire la lezione e prendere appunti
                                   - C'è ancora una lezione?
                                     - No
                                       - Si
                                         - Ho domande?
                                           - No
                                             - Sono sicuro?
                                               - No
                                                 - Si
                                                   - Domandare
                                                     - No
                                                       - Si
                                                         - Prepararsi per la prova scritta
                                                           - Prova scritta (iscriversi all'esame)
                                                             - No
                                                               - Si
                                                                 - Continuare a programmare!
                                                                 - Si
                                                                   - Ho passato la prova laboratorio?
                                                                     - No
                                                                                   - Si
                                                                                     - Prova laboratorio/orale
                                                                                       - Si
                                                                                         - Ho passato la prova laboratorio?
FLOWCHART RESOURCES

https://en.wikipedia.org/wiki/Flowchart
https://www.draw.io
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.

Problem solving

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. **Learn to Use a Debugger:** A debugger will allow you to step line by line through a piece of code (not now)

5. **Seek out More Sources:** tutorials, examples, books, man. Program code ALWAYS with handbook and/or Internet. (NOW)
WHY C
HOW MANY SPOKEN LANGUAGES

~7139

https://www.ethnologue.com/guides/how-many-languages
HOW MANY PROGRAMMING LANGUAGES?

- Wikipedia has a list of 700 programming languages

- Tiobe has a list of 250 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.
  - Their current estimate for **active general purpose programming languages** is between 500 and 2,000.
  - Their current estimate for **all active computer languages** is between 5,000 and 25,000.
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

<table>
<thead>
<tr>
<th>Sep 2021</th>
<th>Sep 2020</th>
<th>Change</th>
<th>Programming Language</th>
<th>Ratings</th>
<th>Change</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>1</td>
<td></td>
<td>C</td>
<td>11.83%</td>
<td>-4.12%</td>
</tr>
<tr>
<td>2</td>
<td>3</td>
<td>↑</td>
<td>Python</td>
<td>11.67%</td>
<td>+1.20%</td>
</tr>
<tr>
<td>3</td>
<td>2</td>
<td>↓</td>
<td>Java</td>
<td>11.12%</td>
<td>-2.37%</td>
</tr>
<tr>
<td>4</td>
<td>4</td>
<td></td>
<td>C++</td>
<td>7.13%</td>
<td>+0.01%</td>
</tr>
<tr>
<td>5</td>
<td>5</td>
<td></td>
<td>C#</td>
<td>5.78%</td>
<td>+1.20%</td>
</tr>
<tr>
<td>6</td>
<td>6</td>
<td></td>
<td>Visual Basic</td>
<td>4.62%</td>
<td>+0.50%</td>
</tr>
<tr>
<td>7</td>
<td>7</td>
<td></td>
<td>JavaScript</td>
<td>2.55%</td>
<td>+0.01%</td>
</tr>
<tr>
<td>8</td>
<td>14</td>
<td>↑</td>
<td>Assembly language</td>
<td>2.42%</td>
<td>+1.12%</td>
</tr>
<tr>
<td>9</td>
<td>8</td>
<td>↓</td>
<td>PHP</td>
<td>1.85%</td>
<td>-0.64%</td>
</tr>
<tr>
<td>10</td>
<td>10</td>
<td></td>
<td>SQL</td>
<td>1.80%</td>
<td>+0.04%</td>
</tr>
</tbody>
</table>

https://www.tiobe.com/tiobe-index/
# C Performance

<table>
<thead>
<tr>
<th></th>
<th>Sudoku solving (CPU sec)</th>
<th>Matrix multiplication (CPU sec)</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Compiled</strong></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Clang:C</td>
<td>1</td>
<td>2.3</td>
</tr>
<tr>
<td>GCC:C</td>
<td>1</td>
<td>2.3</td>
</tr>
<tr>
<td>ICC:C</td>
<td>1</td>
<td>1.8</td>
</tr>
<tr>
<td>GNU:Go</td>
<td>3.8</td>
<td>8.9</td>
</tr>
<tr>
<td>Java:Java</td>
<td>1.7</td>
<td>2.6</td>
</tr>
<tr>
<td>VB:JS</td>
<td>3.7</td>
<td>2.6</td>
</tr>
<tr>
<td>JaegerMonkey:JS</td>
<td>18.1</td>
<td>16.4</td>
</tr>
<tr>
<td>Lua:Lua</td>
<td>50.5</td>
<td>68.3</td>
</tr>
<tr>
<td>llvm-lua:Lua</td>
<td>26.9</td>
<td>31.1</td>
</tr>
<tr>
<td>LuaJIT-JITon:Lua</td>
<td>6.8</td>
<td>2.2</td>
</tr>
<tr>
<td>LuaJIT-JIToff:Lua</td>
<td>16.2</td>
<td>20.8</td>
</tr>
<tr>
<td>Perl:Perl</td>
<td>121.2</td>
<td>230.3</td>
</tr>
<tr>
<td>CPython2:Python</td>
<td>113.9</td>
<td>153.9</td>
</tr>
<tr>
<td>CPython3:Python</td>
<td>119.9</td>
<td>121.9</td>
</tr>
<tr>
<td>IronPython:Python</td>
<td>100.9</td>
<td>202.7</td>
</tr>
<tr>
<td>Jython:Python</td>
<td>136.3</td>
<td>731.4</td>
</tr>
<tr>
<td>PyPy:Python</td>
<td>19.5</td>
<td>8.5</td>
</tr>
<tr>
<td>ShedSkin:Python</td>
<td>4.4</td>
<td>3.7</td>
</tr>
<tr>
<td><strong>Just-in-time</strong></td>
<td></td>
<td></td>
</tr>
<tr>
<td>R:R</td>
<td>98</td>
<td>&gt;1736</td>
</tr>
<tr>
<td>Ruby:Ruby</td>
<td>71.1</td>
<td>628.4</td>
</tr>
<tr>
<td>IronRuby:Ruby</td>
<td>135.5</td>
<td>510</td>
</tr>
<tr>
<td>Jruby:Ruby</td>
<td>&gt;249</td>
<td>238.2</td>
</tr>
<tr>
<td>Rubinius:Ruby</td>
<td></td>
<td>298.1</td>
</tr>
<tr>
<td><strong>Interpreted</strong></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
BITCOIN CORE

- https://bitcoincore.org
- https://github.com/bitcoin/bitcoin

https://github.com/python/cpython
Automated Driving Software Engineer

FCA Fiat Chrysler Automobiles – Auburn Hills, MI

Company Rating
3.4 ★★★★★

Glassdoor Estimated Salary
$87,000/year

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:
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

- Excellent in problem solving
- Strong math and computer graphics experience
- Strong communication skills.
- Strong software engineering skills with experience 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
Lead Programmer (156) (Programming)
Newcastle upon Tyne, United Kingdom - Full-time - REF7601C

JOB DESCRIPTION

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.

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.

Reports To
Producer of the Project

QUALIFICATIONS

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 LANGUAGES I SUGGEST

- Python
- Javascript