

C

Matthias Colin

# C/C++ History

1969 : B

1971 : NB (newB)

1972 : C by Dennis Ritchie and Ken Thompson (NB renamed C)

1978: The C Programming Language, 1st edition

1979 : C with classes by Bjarne Stroustrup

AT&T Bell Labs

1983 : C++

1985 : The C++ Programming Language

1989 : C++ 2.0, The C++ Programming Language, 2nd Edition

C89, 1st standard

1990 : The Annotated C++ Reference Manual

1994 : Standard Template Library (STL)

1998 : C++98, 1st standard

# C/C++ History: standards

1989 : C89 (ANSI C) published C94/95

1990 : ISO/IEC 9899:1990 ; C90 (ANSI C)

1994 : ISO/IEC 9899:1990/Cor.1:1994 ; C94 (correctif)

1995 : ISO/IEC 9899:1990/Amd.1:1995 ; C95

1996 : ISO/IEC 9899:1990/Cor.2:1996 ; C96 (correctif)

1998 : ISO/IEC 14882:1998 ; C++98

1999 : ISO/IEC 9899:1999 ; C99 (plusieurs ajouts du C++, bool, long long)

2001 : ISO/IEC 9899:1999/Cor.1:2001 (correctif)

2003 : ISO/IEC 14882:2003 ; C++03 (98 bug fix)

2004 : ISO/IEC 9899:1999/Cor.2:2004 (correctif)

2011 : ISO/IEC 14882:2011 ; C++11, C++0x

ISO/IEC 9899:2011 ; C11

2012: ISO/IEC 9899:2011/Cor 1:2012 (correctif)

2014 : ISO/IEC 14882:2014 ; C++14, C++1y

2017 : ISO/IEC 14882:2017 ; C++17, C++1z

2018: ISO/IEC 9899:2018 ; C17/18

2020 : ISO/IEC 14882:2020 ; C++20, C++2a

2023 : next one ; C++23, C23

# Compilers

- Families

- GNU C/C++ (gcc, mingw) : gcc, g++
- Microsoft Visual C/C++ : msvc, cl
- LLVM Clang
- NVidia, Embarcadero, Oracle, IBM, Intel, EDG
- **C++ support for each compiler** : [https://en.cppreference.com/w/cpp/compiler\\_support](https://en.cppreference.com/w/cpp/compiler_support)

- IDE

- Visual Studio
- Visual Studio Code (+ léger)
- Eclipse

- Documentation

- <https://fr.cppreference.com>, <https://en.cppreference.com/>
- <https://www.cplusplus.com/reference/>

# Language(s) main usage

- OS
  - Windows
  - Linux
- Drivers
- Embarqué
- Base des autres Langages
  - JVM de Java
  - Framework .NET
  - Python : Interprète et librairies scientifiques et IA
- Jeux Vidéos
- Calcul, performance

# Compilation

Code Source C  
.c, .h

compilateur : gcc, msvc, ...

Code Binaire  
010101010101



# Variables & Types

<https://en.cppreference.com/w/c/language/type>

- int family: headers limits.h, stdint.h
- float family: IEEE754

$$3.75 (10) = 1 \cdot 2^1 + 1 \cdot 2^0 + 1 \cdot 2^{-1} + 1 \cdot 2^{-2}$$

$$= 11.11 (2) = 1.111 \cdot 2^1$$

$$\Rightarrow 0\ 010000000000\ 1110000\dots00$$

$$0.1 (10) = 0.00011001100110011001100\dots$$

- char
  - string.h for char\* or char[ ]: strcpy, strdup, strcat, strcmp, strlen
-

# Standard Library C++ (ex STL)

- includes Library C
  - math.h => cmath
- containers : vector, list, array, set, ...
- I/O
- multi-threadings
- others : everything else
  - random, chrono, exception, memory, meta-programmation



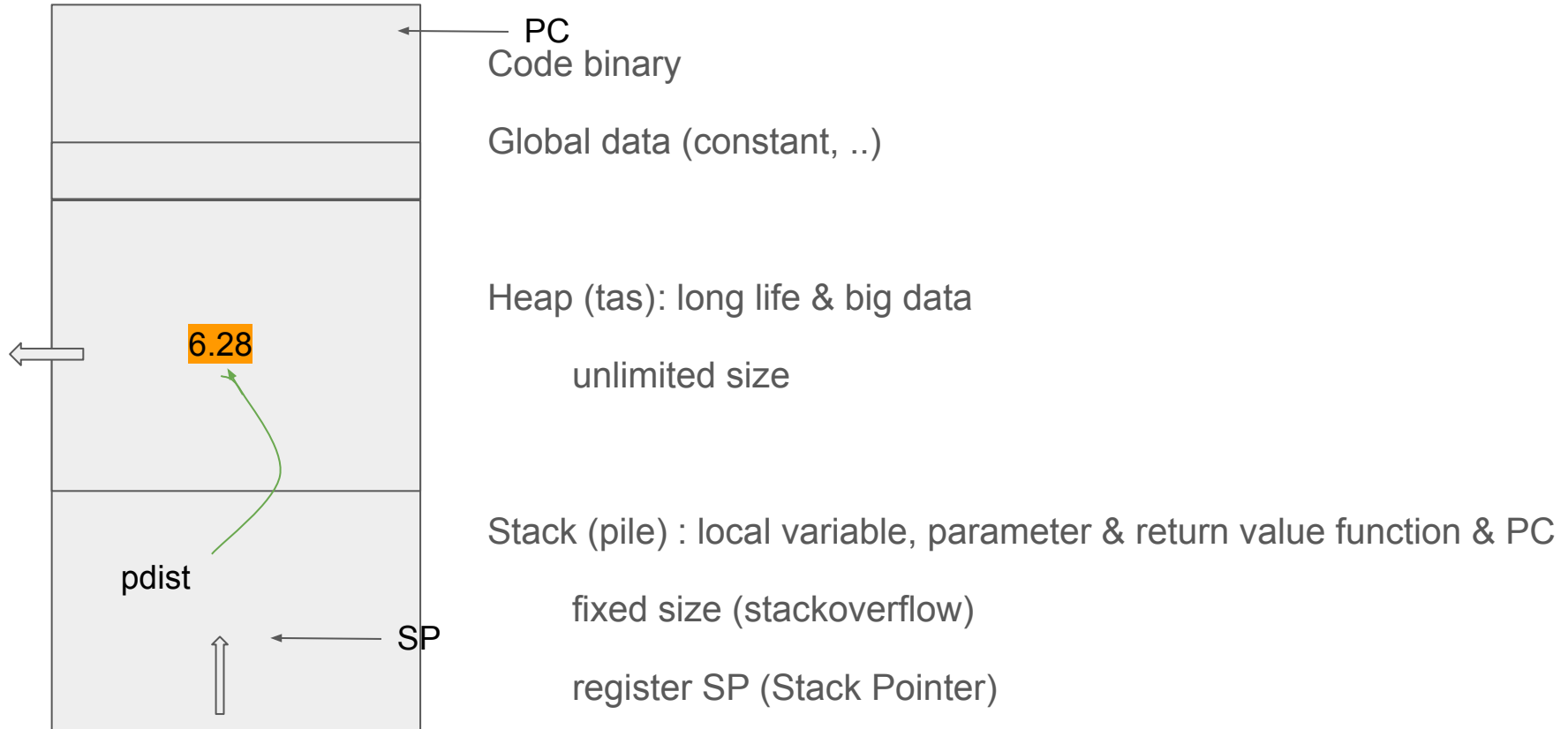
# Operators

- numeric: + - \* / %
  - comparisons: == != < <= > >= <=>
  - logical (and, or, not): && || !
  - bitwise (and, or, xor, not, left shift, right shift): & | ^ ~ << >>
- 
- Article: <https://en.cppreference.com/w/c/language/expressions>

# Control structures

- conditional
  - if ... else
  - switch .... case
- loops
  - for
  - while, do ... while (Warning : infinity loops)

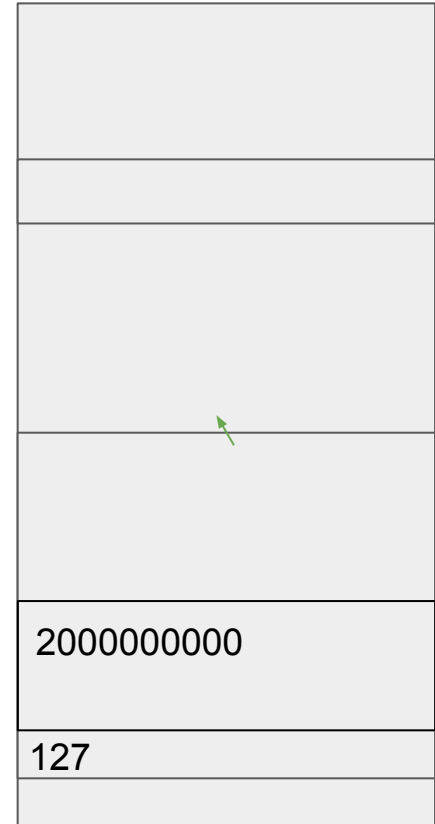
# Memory Management



# Memory : integers

x32: 0x...f0

x8: 0x....f4



# Memory : Exemple Euclide, stack

En rouge, changement de valeur

En vert, dépilement en fin d'appel de fonction

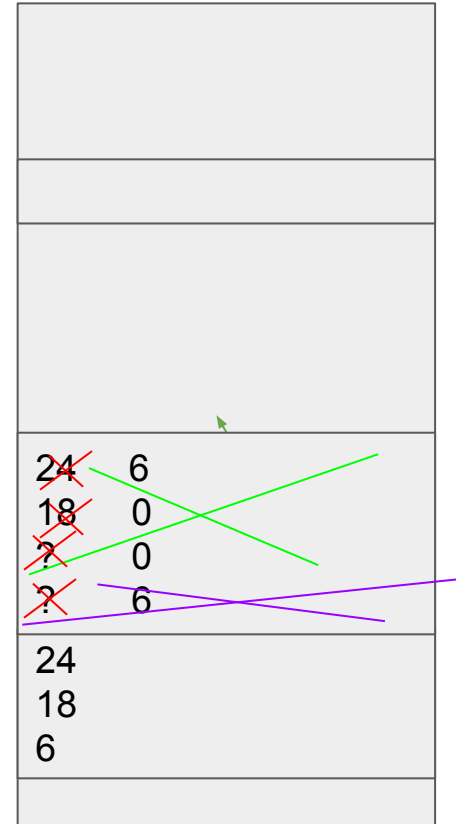
En violet, dépilement valeur de retour de fonction

pgcd

appelEuclide

main

a  
b  
r  
(return value)



# Memory Management (2)

- direct
  - `double dist = 3.14`
- reference to already existing data (C++ only)
  - `double &rdist = dist`
  - use reference normally: `3 + rdist`
- pointer
  - `double * pdist = &dist`
  - `double * pdist = malloc(sizeof(double))` // double in the heap
  - use the data with pointer : `3 + *pdist`
  - `Person * ptr_person = malloc(sizeof(Person(...)))`
  - `(*ptr_person).name` i.e. `ptr_person->name`
- const mode for reference and pointer : data is not modifiable

# Librairies

- librairie dynamique : indépendante, chargée au besoin
  - windows : malib.dll (var env PATH)
  - unix, linux, mac : malib.so (var env LD\_LIBRARY\_PATH)
- librairie statique : à inclure directement dans un un exécutable

# Bibliographie

- Le guide complet du langage C, Claude Delannoy, Eyrolles
- Modern C, Jens Gustedt, Manning (or free pdf)  
<https://gustedt.gitlabpages.inria.fr/modern-c/#org767464e>
-



# Préprocesseur et macros

phase préalable à la compilation en appliquant les macros

code source (.c/.h) => code source enrichi

Macros/directives:

- #define: constantes, fonctions
- #ifdef, #ifndef
- #include
- #if

Gestion des identifiants (variables, fonctions) dynamique