COMMON GROUND

C Programming

université de BORDEAUX

ORGANIZATION

▷ Lectures on Tuesdays

▷ Practical courses on Wednesdays and Thursdays

▷ Exercises to do before lectures

▷ Exercises to do on Moodle (marked)

▷ ⚠Read emails, read the calendar

▷ Communicate via RocketChat (INFO-L2-prog-c) or via email (francois.dross@u-bordeaux.fr)

OBJECTIVES

▷ Strengthen your knowhow with C programing

▷ Know how a C program will be executed - be able to predict the results, notice the mistakes, correct them...

▷ Understand how memory is used in C programs, know how to use it

▷ Know how to use a debugger (GDB)...

C FEATURES

▷ Language using a compiler

▷ Everything needs to be declared before being first used (each variable, constant and expression has a given unique type known at compile time)

▷ Type-safe (type related errors detected at compile time)

▷ Optimized (e.g., once compiled, all information related to types is removed - memory gain)

▷ Middle-level language (1 instruction = more or less constant number of instructions in machine code)

▷ Memory management is explicit (performance gain)

▷ No additional cost at execution time

→ The program only does what you asked for !

A MINIMAL C PROGRAM

```c
/*
    A minimal C program
*/
#include <stdlib.h>
#include <stdio.h>
int main(int argc, char *argv[])
{
    printf("Hello world\n");
    return EXIT_SUCCESS;
}
```

`main`	entry point of the program
`stdlib`, `stdlio`	interface with libraries (e.g. `printf`)
`argc`, `argv`	mechanism for passing arguments
`EXIT_SUCCESS`	program terminaison code

HOW TO COMPILE ? - GCC

▷ GNU Compiler Collection - GCC

▷ Preprocessing, compilation and assembly

```
$>gcc HelloWorld.c -c
```

▷ Linking

```
$>gcc HelloWorld.o -o HelloWorld
```

▷ Execution

```
$>./HelloWorld
hello, world!
```


OPTION SUMMARY

▷ `-std=c99` : determine the language standard

▷ `-Wall` : enables all the warnings

```c
int main(void)
{
  int i, j;
  for(i=0;i<10;++i){
    j+=i;
  }
  return EXIT_SUCCESS;
}
```

```
toto.c:4:5: warning: variable 'j' is
uninitialized when used here
    j+=i;
    ^
toto.c:2:11: note: initialize the
variable 'j' to silence this warning
  int i, j;
           ^
            = 0
1 warning generated.
```


A FEW CODING TIPS

▷ Use a text editor providing syntax assistance

- ▷ indentation
- ▷ syntax highlighting (color)
- ▷ completion of long names

▷ Give priority to readability

- ▷ avoid cryptic writing
- ▷ use evocative names
- ▷ comment without paraphrasing (in English it's better)

▷ Use coding conventions

- ▷ e.g. WebKit (https://webkit.org/code-style-guidelines/) code style guidelines

OBSERVE STANDARDS AND FOLLOW PRACTICES

▷ for instance use spaces wisely :

▷ unary operators, funct. call...

```
p++;
*s
f(a, b)
```

▷ other operators, control...

```
a > b ? a : b
(x * x <= n - 1)
if (p != NULL)
```

▷ or place each statement on its own line

```
x++;
y--;
if (x < y)
    return y - x;
```

▷ but above all, be consistent: choose a style and keep it

▷ and systematically test your code

MEMORY

▷ What do the memory space and a memory address look like ?

C DATA TYPES

▷ What is a data type in C language ?

▷ Examples ?

▷ Properties ?

VARIABLES

▷ What is a variable in C language ?

ARRAYS

▷ What is an array ?

ARRAYS

▷ What does defining an array mean?

▷ How to access to its ith element ?

▷ Is there a way to determine its size ?

EXERCISE

▷ Write a program displaying all the elements of a positive integer array `t` space-separated

ARRAYS

▷ What is the behavior of a program if it accesses to tab[n] where n is the array size ?