



CPPLI : TD 1 : C : Fonction

Nicolas Vansteenkiste Romain Absil Jonas Beleho *
(ESI – HE2B)

Année académique 2019 – 2020

Ce TD¹ aborde les fonctions en langage C dans sa version C11² (ISO/IEC 9899:2011).

Ex. 1.1 Écrivez la fonction de prototype :

```
bool isPrime(unsigned int number);
```

Elle retourne `true`³ ou `false` selon que son argument est premier ou non.
Répartissez prototype et code dans les fichiers `mathesi.h` et `mathesi.c`.

Ex. 1.2 Arrangez-vous pour produire, à l'aide de la fonction `isPrime(unsigned)` de l'Ex. 1.1, la sortie console suivante :

```
Les nombres premiers entre 200 et 349 :  
  
  . . . . .  
  . 211 . . . . .  
  . . . 223 . . . 227 . 229  
  . . . 233 . . . . . 239  
  . 241 . . . . .
```

*Et aussi, lors des années passées : Monica Bastregghi, Stéphan Monbaliu, Anne Rousseau et Moussa Wahid.

1. https://poesi.esi-bru.be/pluginfile.php/1320/mod_folder/content/0/td01_c/td01_c.pdf
2. [https://en.wikipedia.org/wiki/C11_\(C_standard_revision\)](https://en.wikipedia.org/wiki/C11_(C_standard_revision))
3. <https://en.cppreference.com/w/c/types/boolean>

.	251	257	.	.
.	.	.	263	269
.	271	277	.	.
.	281	.	283
.	.	.	293
.	307	.	.
.	311	.	313	.	.	317	.	.
.
.	331	337	.	.
.	347	.	349

Pour la mise en forme, consultez la documentation de l'argument `format`⁴ de la fonction `printf`⁵.

Ex. 1.3 Écrivez la fonction de prototype :

```
void printPrimeFactor(unsigned int number, bool showPower);
```

Elle décompose le nombre non signé passé en paramètre en un produit de facteurs premiers et affiche cette décomposition en notant les puissances ou non avant de passer à la ligne.

Par exemple, avec le nombre 126 en entrée et le paramètre `showPower` `false`, la fonction affiche :

126 = 2 x 3 x 3 x 7

tandis qu'avec le même nombre, mais `showPower` mis à `true`, elle affiche :

126 = 2 x 3² x 7

Répartissez prototype et code dans les mêmes fichiers `mathesi.h` et `mathesi.c` que ceux de l'Ex. 1.1.

Ex. 1.4 Écrivez la fonction de prototype :

```
unsigned gcd(unsigned a, unsigned b);
```

Elle calcule le **plus grand commun diviseur**⁶ (*greatest common divisor*) de `a` et `b` en implémentant l'**algorithme d'Euclide**⁷.

Pour rappel, cet *algorithme récursif* repose sur les deux propriétés suivantes :

$$\begin{aligned}\gcd(a, 0) &= a \\ \gcd(a, b) &= \gcd(b, a \bmod b)\end{aligned}$$

où $a, b \in \mathbb{N}$. Notez que si $a < b$, alors : $\gcd(a, b) = \gcd(b, a \bmod b) = \gcd(b, a)$.

4. <http://www.cplusplus.com/reference/cstdio/printf/>

5. <https://en.cppreference.com/w/c/io/fprintf>

6. https://fr.wikipedia.org/wiki/Plus_grand_commun_diviseur

7. https://fr.wikipedia.org/wiki/Algorithme_d'Euclide

Répartissez prototype et code dans les mêmes fichiers `mathesi.h` et `mathesi.c` que ceux de l'Ex. 1.1.

Ex. 1.5 Produisez, à l'aide de la fonction `gcd(unsigned, unsigned)` de l'Ex. 1.4, la sortie console suivante :

```
gcd(423, 135) = 9 | gcd(423, 130) = 1 | gcd(423, 125) = 1
gcd(426, 135) = 3 | gcd(426, 130) = 2 | gcd(426, 125) = 1
gcd(429, 135) = 3 | gcd(429, 130) = 13 | gcd(429, 125) = 1
gcd(432, 135) = 27 | gcd(432, 130) = 2 | gcd(432, 125) = 1
gcd(435, 135) = 15 | gcd(435, 130) = 5 | gcd(435, 125) = 5
gcd(438, 135) = 3 | gcd(438, 130) = 2 | gcd(438, 125) = 1
```