Algorithmique et Programmation 2
Travaux Pratiques – Séance 3

À déposer à la fin de la présente séance sur Arche

Pour l'ensemble des exercices suivants, pour chaque fonction en `C` demandée, **vous fournirez également les procédures de test associées**.

Indication : afin d'utiliser le type `bool` et les mots-clés `true` et `false` en `C`, il vous suffit d'ajouter la ligne `#include<stdbool.h>` dans l'en-tête de votre fichier source.

Exercice 1 — Le PGCD (ex. 14 page 22)
La fonction `pgcd(a,b : entier naturel) : entier naturel` qui calcule le plus grand diviseur commun entre deux entiers naturels satisfait les axiomes suivants :

[1] `pgcd(0,b) = b`

[2] `pgcd(a,b) = pgcd(b mod a, a)` si $a > 0$

Écrire en `C` une fonction `pgcd(a,b : entier naturel) : entier naturel` qui calcule puis renvoir le `pgcd` de deux entiers naturels en suivant les axiomes ci-dessus. (nom du fichier à déposer sur Arche : `exo1.c`).

Exercice 2 — Le nombre de chiffres (ex. 14 page 22)
La fonction `nb_chiffres(a : entier naturel) : entier naturel` qui calcule le nombre de chiffres dans la représentation décimale de l'entier non nul a satisfait les axiomes suivants :

[1] `nb_chiffres(a) = 1\` si $a < 10$

[2] `nb_chiffres(a) = 1 + nb_chiffres(a/10)` si $a \geq 10$

Écrire en `C` une fonction `nb_chiffres(a : entier naturel) : entier naturel` qui satisfait les axiomes ci-dessus. (nom du fichier à déposer sur Arche : `exo2.c`).

Exercice 3 — Le n^e chiffre (ex. 14 page 22)
La fonction `n_eme_chiffres(n, a : entier naturel) : entier naturel` qui calcule le n^e chiffre dans la représentation décimale de l'entier non nul a satisfait les axiomes suivants :

[1] `n_eme_chiffre(0, a) = a mid 10`

[2] `n_eme_chiffre(n, a) = n_eme_chiffre(n-1, a/10)` si $n \neq 0$

Écrire en `C` une fonction `n_eme_chiffres(n, a : entier naturel) : entier naturel` qui satisfait les axiomes ci-dessus. (nom du fichier à déposer sur Arche : `exo3.c`).

Exercice 4 — Nombre premier (ex. 14 page 22)
Les fonctions `est_premier(n : entier naturel) : bouléen` et `est_premier_FA(d, n : entier naturel) : bouléen` satisfont les axiomes suivants :

[1] `est_premier_FA(d,n) = vraie`, si $d^2 > n$

[2] `est_premier_FA(d,n) = faux`, si $d^2 \leq n$ et d divise n

[3] `est_premier_FA(d,n) = est_premier_FA(d+1,n)`, si $d^2 \leq n$ et d ne divise pas n

[4] `est_premier(n) = est_premier_FA(2,n)`

Écrire en `C` deux fonctions `est_premier(n : entier naturel) : bouléen` et `est_premier_FA(d, n : entier naturel) : bouléen` qui satisfont les axiomes ci-dessus. (nom du fichier à déposer sur Arche : `exo4.c`).

Exercice 5 — La conjecture de Golbach (1742)

La conjecture de Goldbach est l'assertion mathématique non démontrée qui s'énonce comme suit :

> Tout **nombre entier pair** supérieur ou égal à 4 peut s'écrire comme la **somme** de **deux nombres premiers**.

Formulée en 1742 par Christian Goldbach, c'est l'un des plus vieux problèmes non résolus de la théorie des nombres et des mathématiques.

$$
\begin{array}{rcll}
4 & = & 2+2 & \text{(1 solution)} \\
6 & = & 3+3 & \text{(1 solution)} \\
8 & = & 3+5 & \text{(1 solution)} \\
10 & = & 3+7=5+5 & \text{(2 solution)} \\
12 & = & 5+7 & \text{(1 solutions)} \\
14 & = & 3+11=7+7 & \text{(2 solutions)} \\
\vdots & \vdots & \vdots & \vdots \\
50 & = & 19+31=13+37=7+43=3+47 & \text{(4 solutions)}
\end{array}
$$

Écrire en `C` une procédure `sommes_de_Golbach(n : entier naturel)` qui prend en entrée un entier naturel n et affiche à l'écran toutes les solutions $n = p1 + p2$ où $p1$ et $p2$ sont des nombres premiers (nom du fichier à déposer sur Arche : `exo5.c`).