

# Soutien informatique : Langage C. Premiers pas. TP 1-2-3

Sergey Kirgizov, Rose Moskolai

## 1 Édition. Compilation. Exécution.

👉 EXERCICE 1.1. Éditer, compiler et tester le programme suivant :

```
// Compile : gcc -Wall prog_name.c -o prog_name
// Run      : ./prog_name

#include <stdio.h>

int main () {

    printf ("Bonjour le monde ! \n");

    return 0;
}
```

👉 EXERCICE 1.2. Écrire et tester un programme qui imprime la phrase "Bonjour le monde!" 1000 fois.

👉 EXERCICE 1.3. Éditer, compiler et tester le programme suivant. Qu'est-ce qu'il fait ?

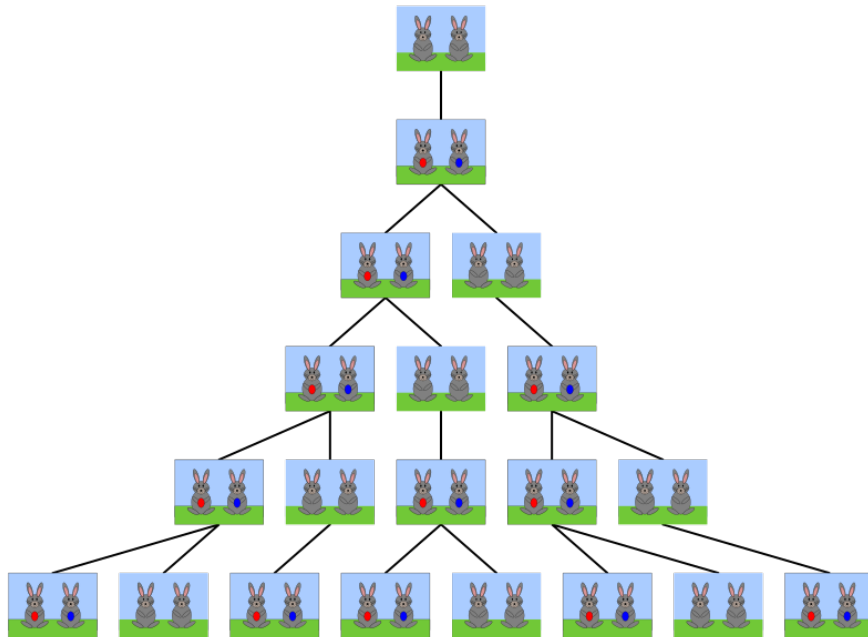
```
#include <stdio.h>

int main () {
    int x,y,t;
    x = 100;
    y = 20;
    printf (" x = %d, y = %d \n", x, y);
    t = x;
    x = y;
    y = t;
    printf (" x = %d, y = %d \n", x, y);

    return 0;
}
```

👉 EXERCICE 1.4. Écrire un programme qui échange les valeurs de deux variables x et y de type int sans utiliser une troisième variable.

## 2 Suites d'entiers



“The number of rabbit pairs form the Fibonacci sequence”  
– MichaelFrey, Sundance Raphael et HB, Wikipedia

👉 EXERCICE 2.1. Les nombres de Fibonacci sont définis par

$$\begin{aligned}a_0 &= 0, \\a_1 &= 1, \\a_n &= a_{n-1} + a_{n-2}.\end{aligned}$$

Écrire un programme qui affiche  $n$ -ème nombre de Fibonacci. Faire deux versions : avec et sans récursion. Laquelle est la plus rapide ?

👉 EXERCICE 2.2. En s’inspirant des deux programmes suivants (avec et sans récursion) qui calculent  $2^n$ , faire deux autres programmes (avec et sans récursion) afin de calculer la valeur de  $k^n$ ,  $k \in \mathbb{N}$ ,  $n \in \mathbb{N}$ .

```
#include <stdio.h>

int puissance_de_2 (int n) {
    if (n == 0) return 1;
    return 2 * puissance_de_2 (n - 1);
}


int main () {
    int k = 8;
    printf ( "2**n = %d \n", puissance_de_2 (k));
    return 0;
}
```

```
#include <stdio.h>
int puissance_de_2 (int n) {
    int i;
```

```

int p = 1;
for (i = 0; i < n; i++) {
    p = p * 2;
}
return p;
}
int main () {
    int k = 8;
    printf ( "2**n = %d \n", puissance_de_2 (k));
    return 0;
}

```


 **EXERCICE 2.3.** Combien de multiplications font les programmes de l'exercice 2.2 pour calculer  $k^n$  ?


 **EXERCICE 2.4.** Voici un algorithme de l'exponentiation rapide :


$$\text{puissance}(k, n) = \begin{cases} k, & \text{si } n = 1, \\ \text{puissance}(k^2, n/2), & \text{si } n \text{ est pair,} \\ k \times \text{puissance}(k^2, (n-1)/2), & \text{si } n \text{ est impair.} \end{cases}$$

Concevoir, écrire et tester un programme qui implémente cet algorithme. Comparer le avec les programmes de l'exercice 2.2.

### 3 Cat fait maison

 **EXERCICE 3.1.** En s'inspirant des programmes qui nous avons vu ensemble en cours, faire un programme qui affiche le contenu d'un fichier.

 **ASTUCE :** Vous pouvez utiliser les fonctions `getc` et `putc` dans un boucle.

 **ASTUCE :** La documentation :

```

man 3 getc
man 3 putc
man 3 stdin
man 3 stdout

```