

## *Feuille de travaux pratiques n°5*

### Nombre variable de paramètres, et fichiers

#### Exercice 5.1 — Nombre variable de paramètres dans un sous-programme

1. Écrivez la procédure `imprimer`, qui écrit sur la sortie standard les différents paramètres qui lui sont passés : chaque paramètre sera écrit sur une ligne nouvelle.  
Cette procédure n’a pas un nombre fixe de paramètres, et aura donc comme paramètre le nombre d’objets à écrire. De plus, cette procédure doit pouvoir manipuler des paramètres de type entier, réel, caractère ou chaîne de caractères. Il faudra donc lui préciser le type des paramètres.

```
imprimer(4, entier, 3, 4, -5, 6);  
imprimer(1, chaine, "Il ne fait pas beau\n");  
imprimer(3, reel, 12.3, 3.5, -8.5);  
imprimer(2, caractere, 'a', 'b');
```

Attention au type du deuxième paramètre (en fait, le dernier fixe).

2. Écrivez le programme permettant de tester cette procédure.
3. Réécrivez la fonction `printf` en ne prenant en compte que `%d`, `%f`, `%c` et `%p`.

#### Exercice 5.2 — Les fichiers

1. Lecture d’une chaîne de caractères.
  - Copiez le programme `lecture.c`. Compilez et exécutez.
  - Écrivez une fonction `myFgets` qui a, à peu près, la même fonctionnalité que la fonction de bibliothèque `fgets`, mais qui renvoie le nombre de caractères effectivement lus sur le flot (au lieu de renvoyer la chaîne lue) et qui en plus “flush” le tampon de l’entrée standard en lisant et en “jetant” les caractères supplémentaires jusqu’au retour-chariot. Le retour-chariot (s’il y en a un) ne sera pas mis dans le tampon.
  - Testez votre fonction par un programme de test adéquat.
2. Création et lecture d’un fichier d’entiers.
  - Copiez le fichier de déclarations `ficEntiers.h`.
  - Écrivez les différents sous-programmes demandés, en fait le fichier `ficEntiers.c`.
  - Testez le tout.
  - Que remarquez-vous quand la lecture du fichier créé se fait octet par octet ?

```
{  
/* autre lecture du fichier créé */  
FILE *fl = ouvrirEnLecture(nomFic);  
int c;  
while ((c = fgetc(fl)) != EOF) {  
    fprintf(stdout, "%c ", c);  
}  
fputc('\n', stdout);  
fermer(fl);  
}
```

- Expliquez.

3. Fichier de structures contenant une chaîne de caractères.
  - Copiez les fichiers `ficChaines.h`, `ficChaines1.c` et `ficChaines2.c`.
  - Compilez le programme `ficChaines1.c`. Exécutez. Le fichier `noms` a été créé, et on ne peut lire son contenu.
  - Compilez le programme `ficChaines2.c`. Exécutez. Pourquoi n'a-t-on pas récupéré les bons noms des enfants ?
  - Écrivez les fichiers `ficChainesbis.h`, `ficChaines1bis.c` et `ficChaines2bis.c` afin de résoudre ce problème.
4. Mise en mémoire des éléments d'un fichier d'entiers. Un fichier contient un nombre illimité d'éléments. Une solution serait de compter le nombre d'éléments du fichier, d'allouer l'espace nécessaire pour le tableau et de transférer les éléments du fichier dans le tableau taillé sur mesure. En fait, le but de l'exercice est de mettre dans un tableau les éléments d'un fichier.
5. Mesurez le temps d'ouverture et de fermeture d'un fichier. Pour ce faire, ouvrez et fermez des milliers de fois des fichiers et chronométrez le temps pris.
6. Mesurez le temps de lecture d'un fichier de 5Mo. Pour ce faire, créez un fichier fictif contenant 5 millions de données, puis lisez le et chronométrez le temps pris.