

Les sous-programmes : devoir en C

Objectifs

- Savoir écrire des sous-programmes ;
- Comprendre les modes de passage de paramètres ;
- Faire la différence entre fonction et procédure ;
- Continuer à appliquer les principes des semaines 1 & 2.

Remarque préliminaire : Pour ce devoir, il est demandé de rendre (de la manière habituelle) un seul fichier qui devra faire apparaître la spécification des sous-programmes et les commentaires liés aux raffinages. En plus, le fichier du programme C devra être envoyé par courriel, en pièce jointe, à cregut@enseeiht.fr.

Remarques complémentaires :

- L'exercice 3 est optionnel. Il faut cependant écrire des sous-programmes de test pour vérifier les sous-programmes écrits dans les exercices 1 et 2.
 - Pour chaque sous-programme, il est important de donner en algorithmique la spécification du sous-programme, c'est-à-dire sa nature (procédure ou fonction), les paramètres et leur mode de passage, le commentaire qui décrit l'objectif et les préconditions. Il ne sera pas nécessaire de remettre ces commentaires dans le programme.
 - Avec l'introduction des sous-programmes, les algorithmes deviennent plus courts et plus simples. En conséquence, il ne sera pas nécessaire de donner les raffinages. Ils doivent toutefois apparaître dans le programme sous la forme de commentaires. Attention, vous devez continuer à commencer par réfléchir sur papier, en faisant des raffinages mais il n'est pas demandé de les rendre dans le devoir.
 - Notons que dans ce devoir, pour les deux premiers exercices, il n'y a pas de saisie utilisateur. Le programme de test (le programme principal) utilise directement les sous-programmes en donnant les paramètres adaptés et affiche les notes ou les élèves. Remarque : il peut être utile de faire un sous-programme pour afficher un élève.
 - Question 2.4. On considère que quand on note un élève on lui donne (la valeur de) sa note et le coefficient.
 - L'exercice 3 propose de constituer une promotion. Elle peut l'être par programme, sans interaction avec l'utilisateur (voir programme sur les factures). Lors de la saisie des notes, nécessairement interactive, on considère que l'utilisateur rend correctement les informations qu'on lui demande. Le programme ne sera donc pas robuste. La question 3.2 est un peu longue. On peut ignorer les statistiques sur la promotion pour la raccourcir un peu.
 - Pour copier une chaîne de caractères, il est conseillé de faire un `strcpy` (`#include <string.h>`).
- ```
strcpy(char destination[], const char source[])
```

On pourra supposer que destination a une capacité suffisante. Cette solution est plus simple mais moins robuste que celle utilisée dans l'exercice sur les factures.

## 1 Gestion des notes d'une promotion d'élèves

L'objectif de ces exercices est d'éditer les notes d'une promotion d'élèves.

### Exercice 1 : Définir le type note

Une note est caractérisée par sa valeur et son coefficient. Par exemple, la note de 14 a été obtenue pour le BE d'algorithmique et programmation qui compte coefficient 1/4.

1.1 Définir le type décrivant une note.

1.2 Écrire un sous-programme qui initialise une note à partir de sa valeur et de son coefficient.

### Exercice 2 : Définir le type Élève

Un élève est caractérisé par son nom, son prénom, son compte d'accès au système informatique, l'ensemble des notes qu'il a obtenues et sa moyenne.

2.1 Définir le type Élève.

2.2 Écrire un sous-programme qui initialise un élève à partir de son nom, son prénom et son compte d'accès au système informatique.

2.3 Écrire un sous-programme qui permet de mettre à jour la moyenne d'un élève.

2.4 Écrire un sous-programme qui permet d'ajouter une note à un élève.

### Exercice 3 : Définir une promotion

Une promotion est un ensemble d'élèves classés par ordre alphabétique de leur nom (et de leur prénom en cas d'homonymie).

3.1 Définir le type Promotion.

3.2 Écrire un sous-programme qui initialise une promotion avec aucun élève.

3.3 Écrire un sous-programme qui édite les résultats de la promotion. Pour chaque élève sont données sa moyenne et, pour le situer par rapport à la promotion, la moyenne générale de la promotion, la plus mauvaise et la meilleure moyenne.

3.4 Écrire un sous-programme qui ajoute un nouvel élève dans une promotion.

3.5 Écrire un sous-programme qui permet d'enregistrer les résultats d'un examen. Le coefficient de l'examen et les notes de chaque élève seront saisies dans le sous-programme.