

IF2B – Examen final

Durée : 1h30

Seule une feuille A4 manuscrite nominative recto-verso est autorisée comme document.

Tous les autres documents ou tout autre dispositif électronique sont interdits.

Barème donné à titre indicatif (±1)

Remarque : les exercices doivent être rédigés en langage C. Le prototype complet des sous-programmes (ou fonctions) doit être fourni. Il n'est pas nécessaire de préciser les « includes ».

Exercices 1 et 2 à rendre sur des copies séparées.

Exercice 1 – Gnome sort (7 points)

Le **gnome sort** est un algorithme de tri dont le principe est simple mais qui est peu efficace. L'objectif de l'exercice est de programmer ce tri pour trier un tableau T d'entiers de taille N.

Le principe de l'algorithme est le suivant : à partir du début du tableau, on regarde si deux éléments successifs sont ordonnés ; s'ils le sont, on passe aux éléments suivants, s'ils ne le sont pas, on les permute et on revient en arrière d'une case (sauf si on est au début du tableau).

Exemple : on veut trier {5, 3, 2, 4}

- {5, 3}, 2, 4} : 5 et 3 ne sont pas dans l'ordre, on les permute et on avance (car on est au début du tableau) – le tableau devient {3, 5, 2, 4}
- {3, 5, 2, 4} : 5 et 2 ne sont pas dans l'ordre, on les permute et on recule – le tableau devient {3, 2, 5, 4}
- {3, 2, 5, 4} : 3 et 2 ne sont pas dans l'ordre, on les permute et on avance (car on est au début du tableau) – le tableau devient {2, 3, 5, 4}
- {2, 3, 5, 4} : 3 et 5 sont dans l'ordre, on avance
- {2, 3, 5, 4} : 5 et 4 ne sont pas dans l'ordre, on les permute et on recule
- {2, 3, 4, 5} : 3 et 4 sont dans l'ordre, on avance
- {2, 3, 4, 5} : 4 et 5 sont dans l'ordre, on a atteint la fin du tableau, on s'arrête, le tableau est trié

Ecrire le sous-programme implémentant l'algorithme du gnome sort prenant en paramètre le tableau d'entiers T de taille N et le triant.

Exercice 2 – Manipulation de fichier (7 points)

L'objectif de cet exercice est de créer un programme capable d'insérer une nouvelle ligne dans un fichier existant.

Le programme se déroule en trois temps :

- On demande à l'utilisateur le contenu de la ligne à insérer et le numéro de ligne où il faudra l'insérer dans le fichier
- On ouvre deux fichiers : un en lecture (le fichier original) et un en écriture (le fichier qui contiendra le texte modifié, qu'on appellera fichier destination)
- Ensuite, on copie chaque ligne du fichier original dans le fichier destination tout en insérant la ligne donnée par l'utilisateur au bon moment

Ecrire en C le programme réalisant l'ensemble de ces opérations.

Exercice 3 – Structures (6 points)

En géométrie :

- Un point dans le plan 2D peut se représenter par deux coordonnées réelles (x, y)
- Une droite dans le plan 2D peut se représenter par une équation cartésienne du type $ax + by + c = 0$ où a , b et c sont des coefficients réels (les points de la droite sont l'ensemble des valeurs (x, y) qui satisfont l'équation).

- 1) D'après la 1^{ère} définition, définir un type structuré permettant de représenter un point dans le plan 2D
- 2) D'après la 2^{ème} définition, définir un type structuré permettant de représenter une droite dans le plan 2D
- 3) Déclarer l'entête d'une fonction qui étant donné deux droites, retourne deux valeurs : un entier qui vaut 0 s'il n'y a pas d'intersection entre les droites ou 1 s'il y a intersection, et un point correspondant aux coordonnées de l'intersection entre les deux droites (le cas échéant) – *vous pouvez utiliser la technique de votre choix pour retourner deux valeurs*
- 4) **BONUS** : Définir le corps de cette fonction

Mémo

Conversion d'une chaîne de caractères en entier : `int atoi(char str[]);`

Conversion d'une chaîne de caractères en flottants : `float atof(char str[]);`

Fonctions usuelles :

`FILE * fopen(char filename[], char accessMode[]);`

`int fclose(FILE * stream);`

`int fprintf(FILE * stream, char format[], ...);`

`int fscanf(FILE * stream, char format[], ...);`

`char * strcpy(char destination[], char source[]);`

`unsigned int strlen(char theString[]);`

`char * fgets(char string[], int maxLength, FILE * stream);`