

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 : vous devez rédiger les algorithmes demandés en langage C. Le prototype (ou entête) complet des sous-programmes doit être fourni. Il n'est pas nécessaire de préciser les « includes ».

Exercice 1 sur une feuille, exercices 2 et 3 sur une autre feuille.

Exercice 1 – Tris, fichiers et structures (10 points)

Un fichier CSV contient une liste d'étudiants et leur note à l'examen d'IF2B. Nous voulons établir un algorithme permettant de réordonner ce fichier en triant les étudiants par ordre alphabétique.

Exemple de contenu du fichier :

Jean,18

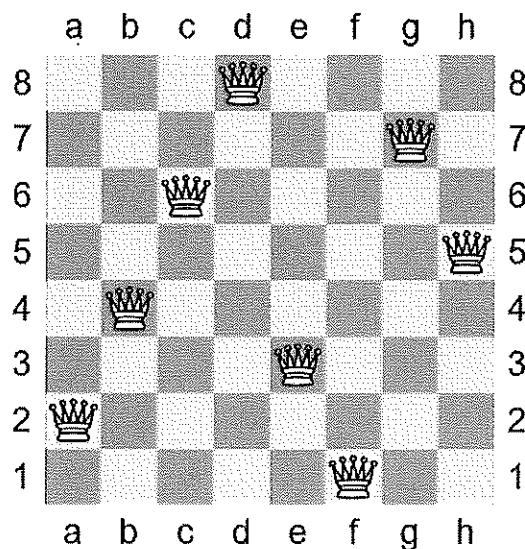
Paul,15

Pierre,0

David,12

- 1) Définir une structure `EtudiantNote` qui contiendra le nom de l'étudiant et sa note
- 2) Définir une fonction `charger_notes` qui prend en paramètre le nom du fichier CSV et initialise un tableau de `EtudiantNote` à partir des données du fichier, la fonction prendra également en paramètre un nombre maximum d'étudiant à lire et retournera le nombre d'étudiants effectivement lus
- 3) Définir une fonction `tri_notes` qui trie le tableau de `EtudiantNote` par nom croissant (algorithme au choix)

Exercice 2 – Tableaux à 2 dimensions (6 points)



Le problème des N dames consiste à placer N dames sur un échiquier sans que les dames ne se menacent : c'est-à-dire que deux dames ne peuvent se trouver sur la même ligne, la même colonne ou la même diagonale. Le nombre N de dames pouvant être placé est égal à la dimension de l'échiquier ($N \times N$).

Dans cet exercice nous représenterons l'échiquier par un tableau à deux dimensions de caractères. Chaque case du tableau contiendra soit le caractère « espace » pour indiquer que la case est vide, soit le caractère « D » pour indiquer que la case contient une dame.

L'objectif de cet exercice est de construire un algorithme capable d'afficher les coordonnées (ligne, colonne) de toutes les cases disponibles pour être occupées par une nouvelle dame sur un échiquier contenant déjà des dames placées.

- 1) Définir *ligne_contient_dame*, qui reçoit en entrée l'échiquier et un numéro de ligne et retourne 1 si la ligne contient une dame, 0 sinon

Pour la suite de l'exercice on supposera qu'une fonction *colonne_contient_dame* similaire est définie, ainsi qu'une fonction *diagonale_contient_dame* qui reçoit en entrée l'échiquier, un numéro de ligne et de colonne et qui retourne 1 si les cases sur la diagonale de cette ligne et de cette colonne contiennent une dame.

- 2) Définir une fonction qui étant donné l'échiquier, affiche l'ensemble des coordonnées (ligne, colonne) sur lesquelles une dame peut être posée (autrement dit l'ensemble des cases qui n'ont pas de dames sur la même ligne, la même colonne ou la même diagonale).

Exercice 3 – Pointeurs (4 points)

Indiquer le plus précisément possible ce qu'affichera le code suivant :

```
#include <stdio.h>

void f(int x) {
    x = x + 2;
    printf("f : %d\n", x);
}

void g(int *x) {
    *x = *x + 2;
    printf("g : %d\n", *x);
}

int h(int *a, int b) {
    *a = *a + b;
    b = b + 1;
    printf("h : %d %d\n", *a, b);
    return *a - b;
}

int main() {
    int a = 3;
    int b = 5;
    int c;

    f(a);
    g(&a);
    c = h(&b, a);
    printf("%d %d %d\n", a, b, c);

    return 0;
}
```

Mémo

Fonction	Description	Exemple d'utilisation
fscanf	Lit des données formatées depuis un fichier, de manière similaire à scanf, mais à partir d'un flux FILE*. Elle retourne le nombre de valeurs correctement lues.	<pre>fscanf(fichier, "%d %s", &age, nom);</pre>
atoi	Convertit une chaîne de caractères en entier de type int. La conversion s'arrête dès qu'un caractère non numérique est rencontré.	<pre>int n = atoi("123");</pre>
atof	Convertit une chaîne de caractères en réel de type float.	<pre>float f = atof("12.3");</pre>
strtok	Découpe une chaîne de caractères en morceaux appelés <i>tokens</i> , selon un ou plusieurs séparateurs. Le premier appel reçoit la chaîne, les suivants reçoivent NULL.	<pre>token = strtok(texte, ";"); puis token = strtok(NULL, ";");</pre>
strcmp	Compare deux chaînes de caractères. Retourne 0 si elles sont identiques, une valeur négative ou positive selon l'ordre lexicographique (négative = 1 ^{ère} chaîne inférieure à la 2 ^{ème} , positive = 1 ^{ère} chaîne supérieure à la 2 ^{ème}).	<pre>if (strcmp(mot, "fin") == 0) { ... }</pre>
strlen	Calcule la longueur d'une chaîne de caractères, sans compter le caractère final '\0'.	<pre>int taille = strlen("Bonjour");</pre>

