

Tp noté

Programmation avancée

14-18 décembre 2020

Exercice 1. Fonctions variadiques.

1. Écrivez une fonction **MaxVal** qui renvoie le maximum de n valeurs réelles passées en paramètre.
2. Testez votre fonction dans un programme principal.

Exercice 2. Pointeur de fonction.

1. Écrivez une fonction **Affiche1** qui prend en paramètres un tableau d'entier T , sa dimension n et qui affichera d'abord le contenu des cases paires (i.e. $T[0]$, $T[2]$, ...) et ensuite les cases impaires.
2. Écrivez une fonction **Affiche2** qui prend en paramètres un tableau d'entier T , sa dimension n et qui affichera la première case $T[0]$, la dernière $T[n-1]$, la seconde $T[1]$, etc...
3. Écrivez une fonction **AfficheTab** qui prendra pour paramètre un tableau d'entiers T , un entier n le nombre d'éléments du tableau et un pointeur de fonctions qui prendra les mêmes arguments. Si le pointeur de fonction a la valeur NULL, la fonction affichera le contenu du tableau en partant de la dernière case. Vous testerez votre fonction d'affichage sur plusieurs tableaux avec les fonctions définies dans les questions 1 et 2.

Exercice 3. Manipulation de fichiers.

Dans cet exercice, on vous demande de mettre au point une base de données clients d'un magasin. La base de données sera stockée dans un fichier. Chaque ligne représentera un client. Pour chaque client, on voudra stocker dans cet ordre son prénom, son nom, sa date de naissance (formatée jj/mm/aaaa), son adresse mail et le nombre d'achats qu'il a effectué. Chaque information sera séparée par un ";" . Voici un exemple :

```
prénomClient1;nomClient1;25/03/1995;mailClient1@gmail.com;0
```

```
prénomClient2;nomClient2;12/12/1982;mailClient2@orange.fr;12
```

```
...
```

1. Écrivez un programme "ajoutClient" qui prendra en paramètre (avec argv) chacune des informations client et les enregistrera dans un fichier nommé "basseClient.txt". Par exemple, la commande `./ajoutClient Jean Dupont 25/12/1985 toto.jean@gmail.com` ajoutera la ligne `Jean;Dupont;25/12/1985;toto.jean@gmail.com;0` dans le fichier "basseClient.txt". Le nombre d'achats sera automatiquement initialisé à 0. Attention, on ne voudra pas écraser les clients déjà enregistrés!
2. Écrivez un programme "incAchat" qui prendra en paramètre le nom et le prénom d'un client et qui lui incrémentera son nombre d'achats. Si deux clients ont le même nom et le même prénom, le programme demandera la date de naissance (scanf) pour les départager (on supposera que deux clients qui ont le même nom et le même prénom ont forcément une date de naissance différente).

3. Écrivez un programme "suppClient" qui prend en paramètre (avec argv) le nom et le prénom d'un client et qui le supprime s'il est dans la base. Si le client n'existe pas, le programme doit renvoyer le message "Le client XXX n'existe pas". Si deux clients ont le même nom et le même prénom, le programme demandera la date de naissance (scanf) pour les départager (on supposera que deux clients qui ont le même nom et le même prénom ont forcément une date de naissance différente).
4. Écrivez un programme "selectClient" qui affichera tous les clients selon un critère particulier. Pour cela, la programme prendra en paramètre (argv), le type de critère et le critère. Concrètement, la commande `./selectClient nom TOTO` devra afficher tous les clients qui se nomment "TOTO" de même, `./selectClient nombreAchat 3` devra renvoyer tous les clients qui ont fait au moins 3 achats. Codez cette fonction pour les critères "nom", "prenom", "dataNaissance", "mail" et "nombreAchat". Si aucun client ne répond au critère, le message "No client found" sera envoyé.
5. Parfois, il arrive qu'on ne connaisse pas le mail du client et/ou sa date de naissance. Pour cela, écrivez le programme "ajoutClient2" qui fait exactement la même chose que "ajoutClient" mais qui comprendra que le caractère "?" signifie que vous ne connaissez pas l'information. Si une information est inconnue, le programme doit inscrire "NONE" dans le champs correspondant. Concrètement, la commande `./ajoutClient Dupont Jean ? toto.jean@gmail.com` ajoutera la ligne `Dupont;Jean;NONE;toto.jean@gmail.com;0` dans le fichier "basseClient.txt". Attention, le "?" n'est possible que pour le mail ou pour la date de naissance. Si vous tapez par exemple la commande `./ajoutClient ? Jean 25/12/1985 toto.jean@gmail.com`, le programme devra renvoyer le message "Le nom doit être fourni". Le message retourné devra s'adapter à l'information manquante.

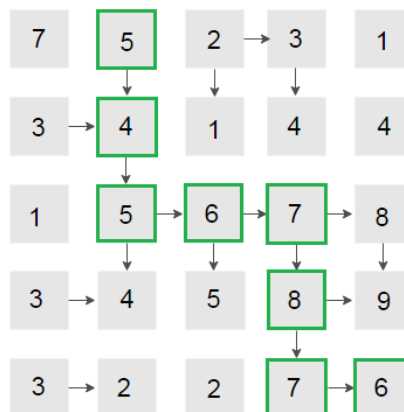
Exercice 4. Plus long chemin.

Soit une matrice M carrée de taille N . On cherche à imprimer la plus longue suite de nombres qui respecte un chemin défini de la manière suivante :

Pour chaque $M(i,j)$, on peut accéder à seulement deux cases, celle de la droite : $M(i,j+1)$ ou celle du bas $M(i+1,j)$ à conditions que le nombre stocké dans ces cases soit égale au $M(i,j) \pm 1$.

Exemple :

Pour la matrice ci-dessous la plus grande suite est : 5,4,5,6,7,8,7,6. Elle est de taille 8.



Écrivez une fonction **RechercheSuite** qui prend en paramètre une Matrice dynamique M , sa taille N et qui revoie la plus grande suite de nombre dans M . On peut démarrer depuis n'importe qu'elle case de la première ligne. (On pourra tester tous les chemins et renvoyer le chemin le plus long).

Exercice 5. Unions.

Le but de cet exercice est de créer un programme qui permet de convertir un nombre de base 50 en base 10. Le cahier des charges est le suivant :

- Le nombre en base 50 sera passé en argument du programme et sera composé de trois chiffres (un chiffre est donc compris entre "00" et "49") ;
- Chaque chiffre en base 50 sera écrit sur 2 caractères (par exemple, le chiffre 0 sera écrit "00") ;
- Le programme devra vérifier que chaque chiffre est bien compris entre "00" et "49" et que le nombre est bien composé de trois chiffres (donc le nombre sera compris entre "000000" et "494949") ;
- Votre programme utilisera une union pour extraire les séquences de 2 chiffres (par exemple, le nombre "120239" sera découpé en "12", "02" et "39" ;
- Le programme affichera le nombre converti en base 10 sur le terminal ;
- Par exemple, la commande `./conversionBase50 120239` devra renvoyer "30139".