TP Programmation C/C++

Les chaînes de caractères et les tableaux

 tv <tvaira@free.fr> v.1.1

Sommaire

Les objectifs de ce tp sont de comprendre et mettre en oeuvre des variables de type tableau ainsi que la manipulation des chaînes de caractères. Certains exercices sont extraits du site www.france-ioi.org.

Les critères d'évaluation de ce TP sont :

- le minimum attendu : on doit pouvoir fabriquer un exécutable et le lancer !
- le programme doit énoncer ce qu'il fait et faire ce qu'il énonce !
- le respect des noms de fichiers
- le nommage des variables ainsi que leurs types, l'utilisation de constantes
- le code est fonctionnel
- la simplicité du code

Les chaînes de caractères : À la bibliothèque

Votre voyage au sein de la caravane des marchands vous a amené de ville en ville, et les ingrédients nécessaires à la fabrication de l'onguent sont achetés un par un. Un ingrédient, cependant, semble poser problème : la seule boutique dans le pays qui vendait cette plante est désormais à court de stock et personne ne sait où il est possible d'en trouver ! En effet, la dernière fois qu'un voyageur en a trouvé, c'était il y a plus d'un siècle ; et depuis, la boutique écoulait le stock tout doucement.

Comme l'ingrédient est absolument nécessaire pour la recette, vous décidez d'aller à la bibliothèque, consulter les archives municipales, afin de voir quelles informations peuvent s'y trouver. On dit en effet qu'un livre spécial contiendrait des informations sur où pousse cette plante ...

En attendant, vos amis marchands vont continuer leur périple ; ils reviendront vous chercher dans quelques semaines.

☞ Objectif : vous allez manipuler des **chaînes de caractères**, c'est-à-dire du texte. Vous apprendrez à manipuler des lignes entières, des mots ou des caractères individuels :

Impression d'étiquettes

Les sous-sols de la bibliothèque municipale sont remplis de milliers de cartons d'archives. Afin d'éviter de passer leurs journées avec la tête tournée à 90 degrés pour pouvoir lire ce qui est écrit sur ces cartons, les bibliothécaires ont adopté un système d'étiquettes où les mots sont écrits de haut en bas avec une seule lettre par ligne.

Étant donné un texte écrit normalement, sur une seule ligne, vous devez afficher l'étiquette correspondante, avec un seul caractère par ligne.

☞ Ce que doit faire votre programme :

La ligne de texte contiendra toujours moins de 50 caractères. On saisira une seule ligne de texte et les caractères du texte seront affichés verticalement. On ne tiendra pas compte des espaces.

☞ Exemples :

```
$ ./impression-etiquettes
Don Quichotte
D
o
n
Q
u
i
c
h
o
t
t
e
```

Question 1. Écrire le programme `impression-etiquettes.c`. Tester et valider.

Écriture en miroir

Alors que vous parcourez de très vieux livres, à la recherche d'indications sur le livre qui vous intéresse en particulier, vous tombez sur un langage que vous ne connaissez pas !

À y regarder de plus près, il s'agit des mêmes mots que ceux que vous utilisez tous les jours, mais tout le texte est écrit "en miroir" : toutes les lettres sont écrites de droite à gauche.

Bien que vous arriviez à déchiffrer les textes présents dans ces livres, cela vous prend beaucoup de temps et vous fatigue beaucoup. Vous décidez d'écrire un programme pour remettre automatiquement dans l'ordre les textes.

☞ Ce que doit faire votre programme :

Il faut tout d'abord lire un entier `nbLignes`, le nombre de lignes du texte et les `nbLignes` suivantes contiennent chacune une ligne de texte qu'il faut inverser. Chaque ligne de texte contient moins de 1000 caractères. Pour chaque ligne du texte original, vous devez l'afficher de manière inversée.

☞ Exemples :

```
$ ./ecriture-miroir
2
tniop a ritrap tuaf li riruoc ed tres en neiR
egangiomet nu tnos ne eutroT al te erveiL eL

Rien ne sert de courir il faut partir a point
Le Lievre et la Tortue en sont un temoignage
```

Question 2. Écrire le programme `ecriture-miroir.c`. Tester et valider.

Petites fiches et gros travail

Afin de faciliter la recherche d'un livre particulier, la bibliothèque municipale s'est dotée d'un système très perfectionné, à base de fiches en carton. Malheureusement, un bibliothécaire stagiaire s'est trompé dans la création des fiches et, à chaque fois, il a oublié des letrres dans le nom de l'auteur !

Votre travail consiste à corriger le contenu des fiches.

☞ Ce que doit faire votre programme :

Le programme lit le nombre de fiches à corriger puis pour chaque fiche : une ligne contenant le nom de l'auteur, puis le caractère et la position (numérotée à partir de 1) où il doit être inséré. Les noms d'auteurs font toujours moins de 64 caractères de long.

☞ Exemples :

```
$ ./fiches-bibliotheque
2
Gorge ORWELL
e 2
George ORWELL

Isac ASIMOV
a 4
Isaac ASIMOV
```

Question 3. Écrire le programme `fiches-bibliotheque.c`. Tester et valider.

Les tableaux : Voyageur de commerce

Le fameux onguent de l'université étant une très grande découverte scientifique, vous décidez d'aider les chercheurs qui le fabriquent. Ceci vous amènera à voyager en compagnie de marchands afin de collecter les différents ingrédients nécessaires à la fabrication de l'onguent.

☞ Objectif : vous allez découvrir la notion de **tableau**. Vous verrez notamment comment manipuler un tableau et quel est son intérêt dans de nombreuses situations :

– Liste de courses : page 4

Liste de courses

Les stocks des ingrédients nécessaires à la réalisation de l'onguent commencent à se vider et les savants vous chargent d'aller en ville acheter une certaine quantité de chaque ingrédient, afin de pouvoir continuer la production pendant le prochain mois. Le comptable étant particulièrement pointilleux, il vous donnera exactement la quantité d'argent dont vous avez besoin, pas une pièce de plus. Heureusement vous savez à l'avance le prix de chaque ingrédient et la quantité dont vous avez besoin.

☞ Ce que doit faire votre programme :

Il y a 10 ingrédients et ils ont tous un prix (entier) au kilo différent : 9, 5, 12, 15, 7, 42, 13, 10, 1 et 20. Votre programme devra lire 10 entiers, le poids (en kilogrammes) qu'il faut acheter pour chaque ingrédient. Il devra calculer le coût total de ces achats.

☞ Exemples :

```
$ ./liste-courses
1
1
1
1
1
1
1
1
1
1

134
```

Question 4. Écrire le programme `liste-courses.c`.

Grand inventaire

Vous êtes dans la boutique du plus grand marchand de la ville, à la recherche d'un certain nombre d'ingrédients. Malheureusement pour vous, c'est la période du grand inventaire et cela peut durer très longtemps ! Vous décidez de les aider. À partir du livre de comptes, sur lequel sont indiqués toutes les ventes et achats de chaque produit, vous allez pouvoir rapidement vérifier si les quantités restantes dans les étalages sont bien les bonnes et s'il n'y a pas eu de vols.

☞ Ce que doit faire votre programme :

Un livre de comptes décrit les achats et ventes successives de produits numérotés de `1` à `n`. Le livre décrit les opérations depuis une situation où le stock de chacun des produits était de zéro.

Chaque ligne du livre de comptes décrit l'achat (augmentation du stock) ou la vente (réduction du stock) d'une certaine quantité de l'un des produits.

Votre objectif est de déterminer pour chaque produit, la quantité restant dans le stock à l'issue de l'ensemble de ces achats et ventes.

Les deux premiers entiers à lire sont le nombre total de produits différents `nbProduits` et un entier `nbOperations` : le nombre d'opérations décrites dans le livre de comptes.

Suivent ensuite `nbOperations` paires d'entiers, où le premier entier de chaque paire est le numéro de l'ingrédient concerné par l'opération, et le deuxième est la quantité. Si la quantité est négative, l'opération est une vente, et si elle est positive, l'opération est un achat du produit indiqué.

Vous devez afficher la quantité restante pour chacun des produits dans l'ordre de leur numéro, une fois l'ensemble des opérations décrites dans le livre effectuées.

☞ Exemples :

```
$ ./grand-inventaire
10
5
1
100
2
50
1
-50
3
20
2
-10

50
40
20
0
0
0
0
0
0
0
```

✍ Faites bien attention au fait que les produits sont numérotés à partir de `1`, tandis que l'indice d'un tableau commence à `0`.

Question 5. Écrire le programme `grand-inventaire.c`.

Visite de la mine

L'un des produits nécessaires pour la fabrication de l'onguent magique, un minerai très rare, ne se trouve qu'en un seul endroit sur la planète, au fond de la plus vieille mine existante, jadis exploitée par le peuple nain. Désormais seuls quelques uns d'entre eux sont encore sur place, afin de guider les voyageurs (commerçants et touristes) au sein de ce dédale de cavernes et galeries.

Après avoir engagé un guide, il vous mène jusqu'à l'endroit prévu mais un petit désaccord sur le paiement de ses services le pousse à vous laisser sur place, sans aucune chance de retrouver la sortie. Heureusement votre robot à conservé en mémoire la suite des déplacements qui vous ont amené de l'entrée jusqu'à votre position actuelle, il ne vous reste plus qu'à suivre le chemin inverse !

☞ Ce que doit faire votre programme :

Il existe 5 types de déplacements, représentés par 5 entiers différents : aller à gauche (1), aller à droite (2), aller tout droit (3), monter (4) et descendre (5).

✍ *Il est conseillé ici d'utiliser un type énuméré pour le déplacement :*

```
typedef enum
{
    AUCUN = 0, /* non indispensable */
    GAUCHE = 1,
    DROITE,
    DROIT,
    HAUT,
    BAS
} Deplacement;

Deplacement deplacement;

// Un déplacement à gauche
deplacement = GAUCHE;
```

Type enum

Le premier entier à lire est le nombre total de déplacements (1000 au maximum). Ensuite, chaque déplacement (représenté par un entier) est indiqué sur sa propre ligne.

Vous devez afficher la suite des déplacements à faire pour aller de votre position actuelle à la sortie.

☞ Exemple :

```
$ ./visite-mine
6
1
3
2
4
4
5

4
5
5
1
3
2
```

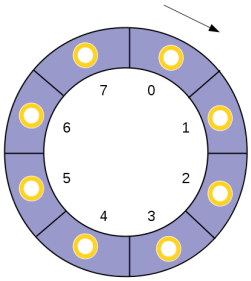
Question 6. Écrire le programme dans le fichier `visite-mine.c`. Tester et valider.

Jeu des anneaux

C'est l'heure de la pause et les enfants font un jeu auquel vous décidez d'inscrire votre robot. Le principe est simple : vous devez déposer huit anneaux sur l'aire de jeu.

☞ Ce que doit faire votre programme :

Votre robot doit partir de la case `0`, déposer les anneaux un par un en se déplacant de gauche à droite en fonction du nombre de cases fourni par un dé à 6 faces (un nombre saisi entre 1 et 6). Il n'y a qu'un seul sens de déplacement.



Votre programme lira un entier compris entre 1 et 6 qui correspondra au déplacement du robot pour déposer un anneau jusqu'à ce que l'aire de jeu soit pleine.

☞ Exemple :

```
$ ./jeu-anneaux
1
01 02 03 04 05 06 07 08
    0
3
01 02 03 04 05 06 07 08
    0         0
4
01 02 03 04 05 06 07 08
 0  0         0
...
```

Question 7. Écrire le programme `jeu-anneaux.c`. Tester et valider.

✍ *L'opérateur* ***modulo %*** *retourne le reste de la division euclidienne. Cet opérateur est très utilisé en informatique (nombre pair ou impair, multiple d'un nombre, intervalle, tampon circulaire ...). Le modulo sur 6 donnera toujours un nombre compris entre 0 et 5 par exemple :*

```
0 % 6 retourne 0
1 % 6 retourne 1
2 % 6 retourne 2
6 % 6 retourne 0
7 % 6 retourne 1
8 % 6 retourne 2
...
```

Bilan

Conclusion : Les types sont au centre de la plupart des notions de programmes corrects, et certaines des techniques de construction de programmes les plus efficaces reposent sur la conception et l'utilisation des types.

Rob Pike (ancien chercheur des Laboratoires Bell et maintenant ingénieur chez Google) :

> « *Règle n°5 : Les données prévalent sur le code. Si vous avez conçu la structure des données appropriée et bien organisé le tout, les algorithmes viendront d'eux-mêmes. La structure des données est le coeur de la programmation, et non pas les algorithmes.* »

Cette règle n°5 est souvent résumée par « Écrivez du code stupide qui utilise des données futées ! ».

