LANGAGE C AVANCÉ

TD 4 : Exercices Divers

Arguments à main, fichiers ...

Arguments de la fonction `main`

1. Écrire un programme qui affiche tous les arguments passés en paramètre en indiquant leur position :

```
$ prog un pomme 1234 toto
1 un
2 pomme
3 1234
4 toto
```

2. Écrire un programme qui additionne tous les nombres passés en argument et qui affiche le résultat.

3. Écrire un programme qui enlève le chemin d'accès d'un nom de fichier (cf. la manpage de basename).

4. Écrire un programme qui prend des nombres en argument et imprime la moyenne de ceux-ci.

5. Similairement, mais si un argument est "`-g`", la moyenne réalisée est la moyenne géométrique. Si l'option "`-c`" est utilisée, des nombres supplémentaires sont demandés au clavier (`fin` permet d'arrêter la saisie).

Fichiers

6. Écrire un programme qui reçoit comme paramètres le nom d'un fichier et un caractère `a`. Le programme compte le nombre d'occurrences du caractère `a` dans le texte. Faire en sorte que, si le nom du fichier est "-", le texte dans lequel on compte le nombre d'occurrences est passé par l'entrée standard (stdin) au programme.

7. Écrire un programme qui reçoit comme paramètres le nom de deux fichiers et deux caractères `a` et `b`. Ce programme copie le premier fichier vers le deuxième fichier, tout en remplaçant les occurrences de `a` par `b`.

8. Comme dans l'exercice 7, mais le programme prend deux chaînes de caractères xx et yy en argument.

9. Écrire un programme qui lit un fichier texte qui contient des données dans le format

```
nom;prénom;adresse;téléphone
```

Écrire une fonction pour chercher une entrée à partir d'un nom fourni par l'utilisateur. Écrire une fonction pour rajouter au fichier une nouvelle entrée. Ajouter des messages de débogage à votre code (par exemple, pendant la recherche d'une donnée). Ces messages sont envoyés vers la sortie stderr. À

l'aide des directives de compilation, écrire un code qui imprime des messages de débogage seulement si cela a été défini pendant la compilation

10. Écrire un programme qui compte le nombre de lignes du fichier passé en argument. Réimplémenter la commande `wc`.

11. Écrire un programme prenant en argument le nom d'un fichier C, et qui vérifie si les accolades ouvertes sont bien fermées

12. Écrire un programme qui supprime les commentaires (`/* */`) dans la source d'un fichier C.