Les documents sont autorisés.

La copie ou les brouillons du voisin ne sont pas des supports autorisés, ni les dispositifs électroniques.
 Éteignez impérativement vos mobiles.

1 Exercice

Question 1 *TP6 myB*

Écrire une commande *cherche repertoire chaîne* qui permet d'obtenir dans le répertoire indiqué l'ensemble des fichiers dont le nom commence par la chaîne indiquée. L'affichage consistera en la simple liste des noms des fichiers correspondants. Ex. :

```
$ cherche /tmp a
af
ag
$
```

Note : on prendra soin d'afficher des messages d'erreurs adéquats, arguments inadéquats, etc

Question 2

Écrire la commande *cherche-rec repertoire chaîne* qui permet d'obtenir la même chose que précédemment mais récursivement dans le répertoire indiqué. La liste obtenue devra être constituée des chemins des fichiers correspondants.

```
$ cherche-rec /tmp a
af
ag
r/ap
tt/zz/afile
$
```

Question 3

Écrire la commande *cherche-long repertoire chaîne* qui permet d'obtenir la même chose que précédemment mais avec un affichage «long» des résultats, c'est-à-dire en accompagnant les noms des fichiers des informations suivantes : droits d'accès (en mode symbolique ie. *rxwxrwxrwx* et taille du fichier.

```
$ cherche-long /tmp a
rwxr-xr-x 23498 af
r--r--r-- 890 ag
rw-r--r-- 4987289 r/ap
-w-r----x 4743879 tt/zz/afile
$
```

2 Exercice

On appelle «mot» une chaîne de caractère du C. On appelle «langage» un ensemble de mots du C.

Question 1

Quels types peuvent représenter mot et langage? Les définir en C...

Question 2

Écrire une fonction `int estDedans(langage *, mot *)` qui permet d'obtenir la valeur C vraie si le mot appartient au langage et la valeur faux sinon.

Question 3

Écrire une fonction `void ajoute(langage *, mot *)` qui permet d'ajouter au langage le mot indiqué s'il ne s'y trouve pas déjà.

Question 4

Écrire une fonction `void enleve(langage *, mot *)` qui permet d'enlever du langage le mot indiqué.

Question 5

Écrire une fonction `langage *union(langage *l1, langage *l2)` qui permet de construire le langage résultat comme union des langages passés en paramètre.

3 Exercice

On souhaite écrire un système de copie de fichier utilisant les tubes nommés ; ce qui permet à un utilisateur d'envoyer des fichiers à un autre utilisateur et de les déposer dans un répertoire qui ne lui appartient pas et sur lequel il n'a pas les droits (par exemple).

Pour envoyer un fichier il faut envoyer dans le tube les données suivantes : le nom du fichier (codée de façon particulière : longueur du nom sur 3 octets + nom), sa longueur (codée sur 9 octets) puis les octets contenus. Le format sera donc :

lg nom			nom		longueur fichier									données				
			...															...

Pour envoyer, par exemple, le fichier de nom ab de longueur 2301 octets, les données qui transiteront dans le tube seront :

lg nom			nom		longueur fichier									données
'0'	'0'	'2'	'a'	'b'	'0'	'0'	'0'	'0'	'0'	'2'	'3'	'0'	'1'	...

'0' représente le code ASCII du caractère 0.

Il n'est pas toujours possible d'envoyer de trop gros fichier par mail, une technique consiste alors à découper le fichier original en plus petits morceaux. On cherche donc à écrire des commandes permettant de découper puis reconstituer un fichier.

Question 1

Écrire la commande `copy tube fichier` qui permet d'envoyer dans le tube indiqué et au format indiqué le fichier spécifié.

Attention : il faudra prendre soin de n'envoyer que le nom du lien, ie. si l'argument spécifié est a/b/c le nom à envoyer est c.

Question 2

Écrire la commande `copy_server tube repertoire` qui permet de recevoir dans le répertoire indiqué les fichiers transmis dans le tube indiqué.

Attention : prendre soin de créer le tube, vérifier qu'il n'existe pas, etc. D'autre part, si on envoie deux fichiers de nom f il doit y avoir un renommage de l'un d'entre eux du côté serveur (à vous de trouver comment les renommer).