

## TP8 - Redirections, droits, mots mêlés

### 1 La partie Linux

**Les droits linux** Récupérer si ce n'est déjà fait le poly : [http://laure.gonnord.org/pro/teaching/AlgoProg1011\\_IMA/unixutil.pdf](http://laure.gonnord.org/pro/teaching/AlgoProg1011_IMA/unixutil.pdf) (dispo à partir de la page du cours).

1. Lire le paragraphe 2.2.2.2 sur la commande `ls`, en particulier la partie sur les *droits*.
2. Quels sont les droits à votre racine ? Quels sont les droits d'un fichier quelconque de votre arborescence ? d'un répertoire quelconque ?
3. Se placer dans `/home/ima3` et vérifier que les droits sont très restrictifs par défaut.
4. Lire le paragraphe 2.2.2.11.
5. Donner accès (`chmod`) à votre répertoire de programmation structurée aux autres membres de votre groupe, de façon à ce que qu'ils puissent copier chez eux vos travaux faits en tp. Faire tester par les autres.

**Redirections (rappel)** Dans la partie suivante, on aura besoin d'un code C déjà écrit, ainsi qu'un fichier `txt` de données.

1. Récupérer les deux fichiers sur la page du cours, avec `wget`
2. Les mettre dans un répertoire TP8.
3. Compiler le fichier C.
4. Ce binaire attend des informations sur le terminal (utilisation de *fgets*). On utilise donc des redirections pour donner le fichier `thegrid.txt` comme entrée :  
`./tp8 <thegrid.txt.`
5. Comment faire pour récupérer les sorties de `tp8` dans un fichier ?

### 2 Le problème des mots mêlés

**Objectifs** Utiliser les tableaux de dimension 2 en C. Conception et implémentation d'algorithmes simples pour les matrices.

**L'énoncé** On va rechercher des motifs dans un tableau 2d de caractères, de taille N fois N. Toutes les cases de ce tableau seront remplies par un caractère minuscule (*i.e.* dont le code ASCII est compris entre 97 et 122). Le but de l'exercice est de savoir si certains mots (appelés *motifs*) sont présents dans la grille, par exemple, le mot **marine** est présent dans la grille suivante (attention on n'a pas rempli toutes les cases) :

		u				e		t	
	m		m						
	m	a	r	i	n	e			u
	t		r		n				
e				i			e		
	u		r		n			m	
		a		t		e		t	
	m		m						e
					e				
e							r		u

Un motif sera représenté par un tableau à une dimension de taille  $N+1$ , avec le marqueur de fin de chaîne `'\0'`. Par exemple, le motif `marine` est représenté par le tableau :

'm'	'a'	'r'	'i'	'n'	'e'	'\0'	?	?	?	?
-----	-----	-----	-----	-----	-----	------	---	---	---	---

On vous fournit un tableau de taille  $20 \times 20$  déjà rempli dans lequel on va rechercher les motifs suivants : `japon`, `marine`, `matin`, `automate`.

### À faire : fichier `tp8.c`

1. Écrire une procédure `affiche_motif` qui prend en paramètre un tableau `motif` et qui imprime le motif correspondant. *On veillera à ne pas afficher le caractère de fin de chaîne `\0`. Tester ie donner une valeur à un tableau `motif`, par exemple `marine`, et imprimer ce tableau.*

Dans un premier temps, on va chercher si le motif est présent à partir d'une case  $(l, c)$ ,  $l$  et  $c$  étant **FIXÉS** à l'avance.

2. Écrire une fonction `cherche_gauche_droite` qui étant donnés  $l$  et  $c$  vérifiant  $0 \leq l \leq N-1$  et  $0 \leq c \leq N-1$ , ainsi que les tableaux de caractères `motif` (à une dimension) et `T` (2d), retourne `true` si le motif apparaît dans `T` horizontalement et de gauche à droite à partir de la case  $(l, c)$ , et `false` sinon. *On fera attention à ne pas accéder à des cases qui n'existent pas, et à ne pas regarder les cases du motif à droite du marqueur de fin `\0`.* Tester : le motif "matin" apparaît dans la grille à partir de la position  $l = 0, c = 0$ .
3. Écrire une fonction `cherche_bas_haut` avec les mêmes arguments et qui retourne `true` si le motif apparaît verticalement et de bas en haut à partir de la case  $(l, c)$ , et `false` sinon. *Mêmes consignes.*
4. Écrire une fonction `cherche_diag1` avec les mêmes arguments et qui retourne `true` si le motif apparaît en "diagonale descendante gauche droite" (l'indice de ligne et l'indice de colonne augmentent) à partir de la case  $(l, c)$ , et `false` sinon. *Mêmes consignes.*
5. Écrire de même rapidement les fonctions `cherche_haut_bas`, `cherche_droite_gauche`, `cherche_diag2` (montante gauche droite), `cherche_diag3` (descendante droite gauche) et `cherche_diag4`.
6. En utilisant les fonctions précédentes, écrire une fonction `trouve_motif` qui prend en paramètre deux tableaux `motif` et `T` et qui cherche si le motif apparaît dans le tableau (il retourne `true` dans ce cas, `false` sinon). Dans le cas positif, on fera imprimer les résultats sous la forme : (résultats **non** conformes au tableau donné)

Le motif marine apparait ...

- en diagonale descendante G/D à partir de la case (1,1)
- horizontalement de gauche à droite à partir de la case (2,1)
- en diagonale montante G/D à partir de la case (8,1)

dans le cas négatif, on pourra se contenter de :

Le motif marine apparait ...

et bien non !

Tester sur les motifs donnés plus haut.

7. Faire en sorte de modifier la matrice d'input (minuscules vers majuscules) lorsqu'un motif a été trouvé.

## La grille

k	m	a	t	i	n	j	t	q	n	p	b	v	u	t	o	l	x	m	q
t	x	m	t	l	n	h	b	y	v	h	i	s	p	h	x	j	s	t	z
f	i	c	b	e	x	p	r	u	d	k	p	d	w	i	o	j	r	p	j
n	w	t	h	o	b	e	x	t	x	y	b	h	a	e	n	z	v	f	t
y	p	j	b	n	t	p	y	l	g	h	a	f	a	j	t	d	n	s	z
n	q	a	w	s	m	a	r	i	n	e	x	i	x	b	v	t	s	t	g
z	c	g	g	c	r	a	i	g	t	h	t	j	h	q	d	n	c	u	m
t	g	l	b	g	m	y	t	f	t	t	g	v	n	o	z	e	p	h	m
i	o	g	r	x	y	p	a	i	o	n	v	y	i	y	e	w	y	f	d
r	k	l	m	x	z	l	d	t	n	q	w	j	y	p	j	y	l	t	a
b	s	x	c	j	x	i	o	v	o	e	m	a	p	y	a	p	l	d	d
g	v	a	s	a	p	b	t	d	w	v	e	o	u	i	q	r	r	r	m
h	v	y	h	p	y	h	e	k	n	f	s	k	f	k	g	w	n	b	z
k	x	g	y	o	q	n	j	j	k	v	q	f	w	a	r	u	j	e	g
w	y	z	h	n	i	p	d	z	q	f	j	n	m	j	h	d	c	n	n
m	p	d	r	r	d	l	i	n	e	q	l	e	r	a	l	d	j	i	e
c	v	p	a	i	a	y	m	d	s	z	p	i	e	h	r	i	u	r	x
a	w	m	e	n	h	r	s	q	h	y	s	d	n	k	l	i	k	a	r
d	y	g	l	d	p	i	n	l	o	m	l	k	y	r	z	f	n	m	w
s	s	q	v	g	c	i	w	n	i	p	q	g	v	d	l	n	l	e	y