

TP 7: Les chaînes de caractères

Guillaume Aubian

Ce document a été grandement inspiré par – et largement copié sur – le document correspondant du responsable précédent de ce cours, Juliusz Chroboczek, avec son accord.

1 Palindromes

Un palindrome est un mot ou une phrase qui se lit de la même façon dans les deux sens. Par exemple, le mot « radar » est un palindrome.

1. Écrivez une fonction

```
int palindrome(const char *s);
```

qui retourne vrai si la chaîne `s` est un palindrome. Vous pouvez supposer que `s` ne contient que des caractères ASCII (dont le code Unicode est inférieur ou égal à 127).

2. Écrivez une fonction `main` qui lit une chaîne au clavier, puis indique s'il s'agissait d'un palindrome. Votre programme ne devra pas se planter même si on lui passe une chaîne très longue — il devra dans ce cas afficher un message d'erreur.

3. Votre fonction donne-t-elle des résultats corrects pour des chaînes contenant des caractères accentués (par exemple « été ») ? (N'essayez pas de corriger le problème, je vous demande juste de le comprendre)

2 ROT-13

ROT-13 est un algorithme simpliste de chiffrage qui est parfois utilisé pour cacher les spoilers et les obscénités sur les listes de discussion. Coder une chaîne en ROT-13 consiste à remplacer chaque caractère alphabétique (minuscule ou majuscule) par le caractère qui est 13 positions plus loin dans l'alphabet, modulo 26 ; les signes de ponctuation et les caractères accentués restent inchangés. Par exemple, la chaîne « Bonjour. » est codée en « Obawbhe. ».

1. Comment fait-on pour décoder une chaîne codée en ROT-13 ?
2. Écrivez une fonction

```
char *rot13(char *dest, size_t dsize, const char *src);
```

qui applique le codage ROT-13 à la chaîne `src` et stocke le résultat dans `dest`, qui peut contenir au plus `dsize` octets. Elle retournera `dest` en cas de succès, et `NULL` si `dest` était trop petit; dans

aucun cas vous ne devrez écrire au-delà de `dest + dsize`. Écrivez une fonction `main` pour tester votre fonction.

3. Modifiez votre fonction `main` pour qu'elle affiche les paramètres de ligne de commande codés en ROT-13. Par exemple, si vous tapez au shell

```
./a.out Bonjour, le C.
```

votre programme affichera

```
Obawbhe, yr P.
```

3 Pendu

Le but de cet exercice est d'implémenter un jeu du pendu.

1. Écrivez une fonction

```
int member(char c, const char *f, int n);
```

qui retourne vrai si le caractère `c` apparaît dans le tableau `f` de longueur `n`.

2. Écrivez une fonction

```
partial_display(const char *s, const char *f, int n);
```

qui affiche la chaîne `s` en remplaçant par des astérisques « `*` » tous les caractères qui ne se trouvent pas dans le tableau `f` de longueur `n`. Elle retournera le nombre d'astérisques qu'elle a affichés. Par exemple, si `s` vaut "toto" et `f` vaut `{t}`, votre fonction affichera « `t*t*` » et retournera 2.

3. Écrivez une fonction `main` qui prend en paramètre de ligne de commande un mot `s` à deviner, puis :

a) lit un caractère, et l'ajoute à un tableau `f` ;

b) affiche `s` en remplaçant les caractères ne se trouvant pas dans `f` par des astérisques ;

c) recommence tant que tous les caractères de `s` ne sont pas dans `f` et que le nombre de tentatives n'a pas dépassé 10. Il faudra bien sûr s'assurer que vous avez alloué suffisamment de place pour le tableau `f`.

4. Modifiez votre programme pour en faire un jeu du pendu complet, avec choix aléatoire d'un mot choisi dans `/usr/share/dict/words` et affichage d'une potence en art ASCII.