

Bases de la Programmation

TP 5 – Fonctions (suite)

V. Padovani, PPS - IRIF

Exercice 1 : Pendu

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

1. Ecrire une fonction `int char_in(char c, char *t, int n)`. Cette fonction suppose que `t` est l'adresse d'un tableau de caractères de longueur au moins `n`. Elle doit renvoyer 1 si le caractère `c` apparaît dans les `n` premières cases du tableau, et 0 sinon¹.
2. Écrire une fonction `int partial_display(char *s, char *t, int n)` affichant une chaîne d'adresse `s` en remplaçant par des astérisques (*) tous les caractères qui ne se trouvent pas dans les `n` premières cases du tableau d'adresse `t`. La fonction renverra le nombre d'astérisques affichés.
Par exemple, si la chaîne est "toto", si `n` vaut 1 et si la première case du tableau contient 't', la fonction affichera "t*t*" et retournera 2.
3. Ecrire une fonction `main` lisant une chaîne à deviner au clavier, puis effectuant le traitement suivant :
 - (a) lire un caractère, l'ajouter à un tableau auxiliaire `tab` suffisamment grand ;
 - (b) afficher la chaîne, en remplaçant par des astérisques tous les caractères non encore ajoutés au tableau `tab` ;
 - (c) recommencer tant que tous les caractères de la chaîne ne sont pas dans `tab` et que le nombre de tentatives n'a pas dépassé 10.

Il faudra bien sûr s'assurer que le tableau `tab` est suffisamment grand.

4. Modifiez votre programme pour en faire un jeu du pendu complet.

Exercice 2 : Master Mind

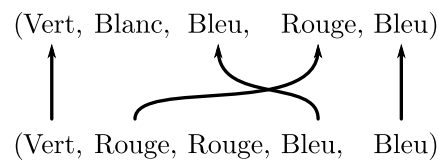
Le jeu de Master Mind se joue à deux joueurs, à l'aide de pions colorés – jaune, bleu, rouge, vert, blanc et noir. Le premier joueur choisit une suite de 5 pions, qui est dissimulée au second. Le but du jeu est, pour le second joueur, de deviner la suite choisie.

A chaque étape du jeu, le second joueur propose une suite de 5 pions. Le premier joueur examine cette suite et, s'il ne s'agit pas encore de la suite choisie, doit lui dire :

- combien de pions sont à une position correcte ;
- parmi les pions restants :
combien peuvent être déplacés sur des positions restantes, correctes et distinctes.

Par exemple la suite à deviner est (Vert, Blanc, Bleu, Rouge, Bleu) et si la suite proposée par le second joueur est (Vert, Rouge, Rouge, Bleu, Bleu) :

1. Une fonction similaire apparaît dans la bibliothèque standard sous le nom `memchr`.



- Le pion *Vert* et l'un des deux pions *Bleu* sont à des positions correctes.
- Parmi les pions restants, l'un des deux pions *Rouge* et le second pion *Bleu* peuvent être déplacés sur des positions restantes, correctes et distinctes.
- La réponse du premier joueur est donc :
2 pion(s) placé(s).
2 pion(s) déplaçable(s).

Ecrire une fonction simulant le rôle du premier joueur : invoquée sur trois tableaux d'entiers, l'un représentant la suite à deviner l'autre la suite proposée, le troisième étant un tableau à deux cases, cette fonction devra inscrire dans le dernier tableau la réponse du premier joueur.

Les couleurs seront représentées par des entiers successifs distincts. La fonction ne devra pas altérer le contenu des deux premiers tableaux. Elle devra être écrite de manière suffisamment flexible pour gérer un nombre arbitraire de couleurs et des suites de longueur quelconque.

S'il vous reste du temps : compléter le programme en un jeu complet, permettant à un utilisateur de jouer contre la machine sur une configuration tirée au hasard par celle-ci.

Indication. Le calcul des réponses peut se faire avec un tableau de plus, un *tableau de marquage* contenant des 0 (position encore utilisable pour un déplacement si elle n'est pas celle d'un pion correctement placé) et des 1 (position déjà utilisée pour un déplacement).