INF 111, L1, 2004/2005

Feuille 6

Octobre 2004

Booléens - Tableaux de caractères

Laure Gonnord
http://www-verimag.imag.fr/~gonnord/

1 Retour sur les booléens

☺ En C, il n'existe pas de type booléen prédéfini, mais si on fait des tests avec des programme du style :

```
int main(void)
{
  if (1)
    {printf("Hop");}
  else
    {printf("Hup");}
  return 0;
}
```

on s'aperçoit que c'est l'entier 0 qui joue le rôle de "faux" et toutes les autres valeurs sont "vrai". Pour simplifier, on va dire que 1 est "vrai".

☺ En C pour des raisons de lisibilité du code, on va définir un nouveau type `bool` qui est en fait un "alias" pour le type entier. On définit ensuite les constantes `VRAI` et `FAUX`.

```
#define bool int

#define VRAI 1
#define FAUX 0
```

On peut ainsi donner le type `bool` comme type de sortie d'une fonction, ou comme type des variables d'entrées. On pourra retourner `VRAI` ou `FAUX` sans problème.

Utilisation Les variables logiques déclarées par exemple comme cela : `bool b = VRAI` peuvent bien sur être utilisées à l'intérieur d'une expression conditionnelle : `if(b) {...} else {...}` veut dire "si la variable `b` est `VRAI`, ... ".

Remarque 1 On écrit bien `if (b) ...` et pas `if (b == VRAI)`, on va toujours au plus simple.

2 La table ASCII

☺ La table ASCII est une table qui fait la correspondance entre un caractère et son entier qui le code en machine. Pour avoir une telle table, voir à la fin du cahier jaune de TP.

Remarque 2 On remarquera que $code(Z) = code(z) - 32$.

☺ Exemples d'utilisation :
- Pour obtenir un caractère à partir de son code ASCII, on déclare la variable comme un charactère et on lui donne la valeur voulue. Par exemple, on peut faire `char mon_a = 97;`

- Pour imprimer le caractère désigné par le code ASCII 102, on fait `printf("%c",102)`.
- Pour imprimer le code ASCII du caractère `B`, on fait `printf("%d",'B')`.
- Pour remplir un tableau de caractères désignés par leur code, on déclarera le tableau `char tab[MAX]` par exemple, que l'on pourra remplir à l'aide d'instructions de la forme : `tab[i] = 67;`

3 Recherches dans un tableau de caractères

(D'après un TD de Michaël Périn)

1. Écrire un programme qui déclare un tableau `tab` de 500 caractères (entre 'a' et 'z', minuscules), et qui l'initialise avec des caractères tirés au hasard. On utilisera la fonction `int hasard(int binf, int  bsup)` qu'on ne redemande pas de coder, et on utilisera la table ASCII de la fin du cahier jaune de TP.
2. Modifier le programme pour qu'il affiche chaque caractère du tableau suivi de son code ASCII ainsi que la majuscule correspondante.

   ```
   c[0] = 'b' code = 98 maj = 'B'
   c[1] = ...
   ```
3. Afficher la plus petite lettre (pour l'ordre alphabétique) ayant été tirée :

   ```
   La plus petite lettre est : 'b'.
   ```
4. Demander une lettre à l'utilisateur de taper une lettre et afficher les indices des cases où elle se trouve. On utilisera `getchar()` :

   ```
   On trouve la lettre 'c' dans les cases 35, 196, 354, 488.
   ```
5. Déterminer et afficher le nombre d'apparitions de la lettre donnée par l'utilisateur. On utilisera `getchar()` :

   ```
   On trouve la lettre 'c' 4 fois.
   ```
6. Déterminer le premier indice d'apparition de chacune des lettres et afficher sous la forme :

   ```
   'a' n'apparait jamais
   'b' apparait pour la première fois en position 0
   'c' apparait pour la première fois en position 35
   ...
   ```

   On n'utilisera pas de tableau, et pour chaque lettre on utilisera une boucle while.
7. Déclarer et initialiser un tableau `occurrence` qui va nous servir à compter le nombre d'apparitions de chaque lettre. De quelle taille est-il ?
8. Remplir le tableau `occurrences` avec le nombre d'apparition de chaque lettre dans le tableau `tab`, et afficher le résultat ensuite sous la forme :

   ```
   'a' apparait 0 fois
   'b' apparait 1 fois
   'c' apparait 4 fois
   ...
   ```
9. Afficher le contenu du tableau `tab` dans l'ordre alphabétique. On utilisera le tableau `occurrences`, et si le caractère `'a'` n'apparaît pas, on n'affichera pas `'a`.
10. Utiliser le tableau `occurrences` pour modifier le tableau `tab` afin qu'il soit trié par ordre alphabétique, puis afficher le tableau `tab` afin de vérifier qu'il est trié. Indication : avec le tableau `occurrences`, on connaît le nombre d'occurrences de chaque caractères. On peut donc écraser le contenu du tableau `tab` sans perdre la moindre donnée.

4 En TP

On commencera par finir ce TD, puis on fera le TP sur la propagation d'une nouvelle : cahier jaune de TP pages 55 à 57.