

Feuille 6 : Chaîne de caractères

Variables de type char : rappels

On rappelle que le code ASCII permet d'identifier un caractère avec son code (entier entre 0 et 255). Pour l'affichage, on rappelle que si on a `char a='b'`; (i.e. la variable `a`, de type `char`, contient le caractère `'b'`), `printf('a vaut %c',a)`; provoquera l'affichage `a vaut b` alors que `printf('a vaut %d',a)` provoquera l'affichage `a vaut 98`, le code ASCII du caractère `'b'` valant 98.

Une conséquence du codage est que l'on peut comparer deux caractères, la comparaison étant celle de leur code. A noter que les codes des lettres minuscules d'une part, majuscules d'autre part, sont consécutifs et classés en respectant l'ordre alphabétique : le code ASCII du caractère `'a'` vaut 97, celui de `'b'` vaut 98, ..., celui de `'z'` vaut 122, celui de `'A'` vaut 65, ... et celui de `'Z'` vaut 90. De façon analogue, les codes des chiffres sont consécutifs et classés suivant l'ordre croissant des valeurs correspondant aux chiffres : le **caractère** `'0'` a pour code ASCII 48, le **caractère** `'1'` a pour code ASCII 49, ... et le **caractère** `'9'` a pour code ASCII 57. Une conséquence pratique est la suivante : dans le cas où on a :

```
char a, b; int d;
a='3' ; b = '0';
d=a-b;
printf('%c-%c=%d',a, b, d);
```

l'affichage sera `3-0=3` car, dans ce cas, `a-b` est la différence entre le code du caractère contenu dans la variable `a` (soit 51 pour `'3'`) et celui du caractère contenu dans la variable `b` (soit 48 pour `'0'`) et cette valeur sera stockée dans la variable `d` de type `int`.

Tout au long de cette feuille, on pourra considérer la fonction `MyGets(char chaine[])` suivante qui lit au clavier une chaîne de caractères :

```
void MyGets(char chaine[])
{ char c;
  int lg=0;
  while( (c=getchar())!='\n')
  {
    chaine[lg] = c;
    lg++ ;
  }
  chaine[lg] = '\0';
}
```

Exercice 1 :

1. Écrire une fonction, nommée `comptchr`, qui retourne le nombre de fois où le caractère contenu dans le paramètre formel `c` apparaît dans une chaîne de caractères.
2. Écrire une fonction, nommée `comptAlphabet`, qui retourne le nombre de lettres (majuscules ou minuscules) dans une chaîne de caractères.
3. Écrire une fonction, nommée `occurrenceMax`, qui retourne le nombre maximum d'occurrences (de présence) d'un caractère dans une chaîne de caractères.

4. Écrire un programme qui :
 - a) saisit une chaîne d'au plus 132 caractères,
 - b) saisit un caractère,
 - c) appelle successivement les trois fonctions précédentes et affiche les résultats obtenus,
 - d) supprime de la chaîne, le caractère saisi et affiche la chaîne obtenue.

Exercice 2 :

1. Écrire une fonction, nommée `comparChaine`, qui prend en paramètre deux chaînes de caractères `s1` et `s2` et qui retourne -1 si `s1 < s2`, 0 si `s1 = s2` et 1 si `s1 > s2` (ne pas utiliser `strcmp`). On utilisera l'ordre lexicographique pour comparer deux chaînes de caractères ce qui correspond, dans le cas de deux mots, au classement dans un dictionnaire.
Exemples : "polaire" < "solaire", "solaire" < "soleil" et "sol" < "solaire".
2. Écrire une fonction, nommée `chaineIncludeInd`, qui prend en paramètre deux chaînes de caractères `s1` et `s2` et un entier `iDebut` (supposé positif et inférieur à la longueur de `s2`) et qui retourne 1 si `s1` est contenue dans `s2` à partir de l'indice `iDebut` exactement, 0 sinon.
Exemple : Prenons `s2 = "Il entend"`. Si `s1 = "en"`, la fonction retournera 1 si `iDebut` vaut 3 ou 6, et 0 sinon. Si `s1 = "end"`, la fonction retournera 1 si `iDebut` vaut 6, et 0 sinon. Si `s1 = "ende"`, la fonction retournera toujours 0.
3. Écrire une fonction, nommée `chaineInclude`, qui prend en paramètre deux chaînes de caractères `s1` et `s2` et qui retourne 1 si la chaîne `s1` est contenue dans la chaîne `s2`, 0 sinon.
4. Tester les fonctions précédentes.

Exercice 3 :

On dit qu'une chaîne de caractères est un palindrome si la succession des caractères est la même quand on la parcourt de gauche à droite ou de droite à gauche. Par exemple, le mot "ressasser" est un palindrome.

Écrire et tester une fonction, nommée `palindrome`, qui retourne 1 si la chaîne de caractères en paramètre est un palindrome, 0 sinon.

Exercice 4 :

1. Écrire une fonction qui convertit une chaîne de caractères formée de chiffres en sa valeur numérique de type `int`.
Indication au niveau algorithmique sur un exemple :
On souhaite convertir la chaîne 2468 ; pour cela, on pourra remarquer que $24 = 2 \times 10 + 4$ puis que $246 = 24 \times 10 + 6$ et enfin que $2468 = 246 \times 10 + 8$ (ce que l'on peut symboliser par l'écriture : $2468 = 8 + 10(6 + 10(4 + 10 \times 2))$).
2. Sachant que `sizeof(int)` vaut 4 (ce qui est vrai sur la plupart des processeurs), combien de chiffres au maximum peut contenir une chaîne afin d'assurer le bon fonctionnement de la fonction précédente ?
3. Écrire un programme qui saisit une chaîne d'au plus 9 chiffres et affiche l'entier correspondant.
4. Ajouter à la fonction du 1. le contrôle de la prise en compte des caractères de la chaîne tant qu'ils sont des chiffres.
5. Ajouter à la fonction du 4. la possibilité que le premier caractère soit le signe de l'entier (entier positif si pas de signe ou si le premier caractère est '+', entier négatif si le premier caractère est '-').