

R101 – TD5

Exercice 1 : chercher un caractère dans une chaîne de caractères

Ecrire la fonction `trouveCarDansStr ()` de profil :

```
fonction trouveCarDansStr (caract : in caractere,  
                           chaine : in string,  
                           debut : in entier_naturel) renvoie entier_naturel;
```

Cette fonction renvoie l'indice de la première occurrence de `caract` dans `Chaine`, à partir de l'indice `debut`. Si `caract` n'a pas été trouvé, la fonction renvoie la taille de `Chaine`.

Exercice 2 : comptage d'un caractère (V1)

Le but de cet exercice est d'afficher le nombre d'occurrences d'un caractère dans une chaîne de caractères. Pour cela :

- écrire la fonction `comptCaract ()` qui compte le nombre d'occurrences d'un caractère passé en paramètre, dans une chaîne de caractères passées également en paramètre ; pour cela, parcourir la chaîne caractère par caractère et incrémenter le nombre d'occurrence quand il le faut ;
- écrire l'algorithme qui teste cette fonction.

Exercice 3 : comptage d'un caractère (V2)

Le but de cet exercice est d'afficher le nombre d'occurrences d'un caractère dans une chaîne de caractères. Pour cela :

- écrire la fonction `comptCaract ()` qui compte le nombre d'occurrences d'un caractère passé en paramètre, dans une chaîne de caractères passées également en paramètre ; pour cela, utiliser la fonction `trouveCarDansStr ()` ;
- utiliser l'algorithme précédent qui teste cette fonction.

Exercice 4 : comptage d'un petit ensemble (V1)

Le but de cet exercice est d'afficher le nombre d'occurrences de chaque élément d'un petit ensemble de valeurs dans une collection. Ici, on comptera le nombre de voyelles contenues dans une chaîne de caractères. Pour cela :

- initialiser les compteurs ;
- pour chaque valeur de l'ensemble (chaque voyelle), balayer complètement la collection (la chaîne de caractères) et compter le nombre d'occurrences de cette valeur (le nombre de voyelles) ;
- calculer le nombre total d'occurrence des éléments de l'ensemble (le nombre total de voyelles) ;
- afficher le nombre d'occurrences de chacun des éléments de l'ensemble (de chacune des voyelles) ainsi que leur fréquence relative (au nombre total d'éléments de l'ensemble (de voyelles)) ;

Remarque :

- on pourra utiliser les fonctions `comptCaract ()` et/ou `trouveCarDansStr ()` ;
- on rangera l'ensemble des voyelles dans une chaîne de caractères constante.

Exercice complémentaire #1 : comptage d'un petit ensemble (V2)

Le but de cet exercice est d'afficher le nombre d'occurrences de chaque élément d'un petit ensemble de valeurs dans une collection. Ici, on comptera le nombre de voyelles contenues dans une chaîne de caractères. Pour cela :

- initialiser les compteurs ;
- pour chaque valeur de la collection (chaque caractère de la chaîne), balayer complètement le petit ensemble (l'ensemble des voyelles) et incrémenter le bon nombre d'occurrence quand il le faut ;
- calculer le nombre total d'occurrence des éléments de l'ensemble (le nombre total de voyelles) ;
- afficher le nombre d'occurrences de chacun des éléments de l'ensemble (de chacune des voyelles) ainsi que leur fréquence relative (au nombre total d'éléments de l'ensemble (de voyelles)) ;

Remarques :

- on pourra utiliser les fonctions `comptCaract()` et/ou `trouveCarDansStr()` ;
- on essaiera de comparer les 2 versions quant à l'efficacité.

Exercice complémentaire #2 : comptage d'un grand ensemble

Dans les exercices précédents, il est évident que plus la taille du petit ensemble est grande, plus la solution qui consiste à parcourir la collection pour chaque élément de l'ensemble devient lourde. Il faudrait donc pouvoir :

- parcourir la collection une seule fois ;
- trouver très rapidement (sans parcours) si l'élément fait partie de l'ensemble ;
- déterminer, à l'aide d'un calcul simple, quel est le compteur à incrémenter.

Ici l'ensemble dont il faut compter le nombre d'occurrences de chacun des éléments est constitué des chiffres (caractères numériques), des lettres majuscules et des lettres minuscules. La collection est toujours une chaîne de caractères.

Il va donc être nécessaire d'avoir 62 compteurs (10 chiffres, 26 majuscules et 26 minuscules), donc un tableau de 62 entiers naturels.

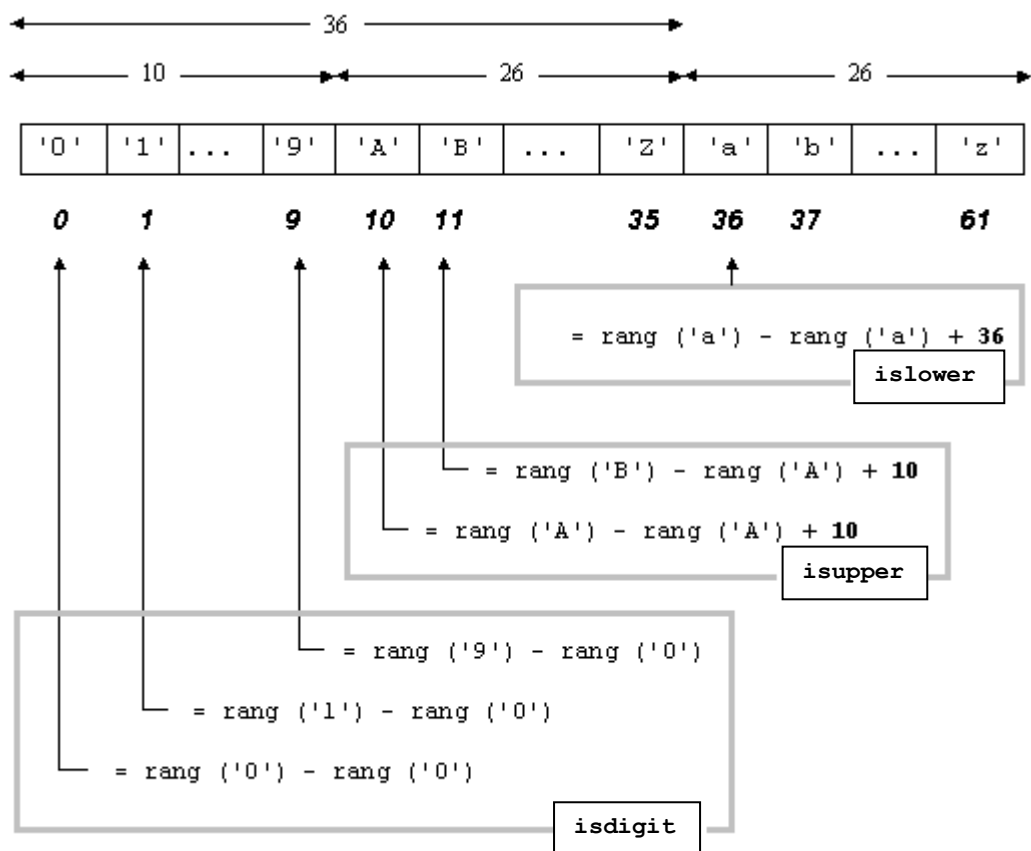
Comment savoir si l'élément (le caractère considéré de la chaîne de caractères) fait partie de l'ensemble ? Trois tests suffisent, c'est un chiffre (`isdigit()`), une majuscule (`isupper()`) ou une minuscule (`islower()`).

Les fonctions `isdigit()`, `isupper()` et `islower()` sont 3 prédicats de profils:

```
fonction isdigit (Caract : in caractere) renvoie boolean; ....
```

Comment savoir quel compteur incrémenter ? Il suffit de savoir comment on organise le tableau des compteurs. Les 10 premiers éléments du tableau serviront à comptabiliser chacun des chiffres. Les 26 éléments suivants serviront pour les majuscules et les 26 derniers pour les minuscules. Donc, pour connaître le compteur à incrémenter l'indice sera :

- si le caractère est un chiffre, le rang de ce chiffre diminué du rang du caractère '0', exemple le caractère est '7', le compteur correspondant est à l'indice $\text{rang}('7') - \text{rang}('0')$, donc à l'indice 7 ;
- si le caractère est une majuscule, le rang de cette majuscule diminué du rang du caractère 'A', exemple le caractère est 'S', le compteur correspondant est à l'indice $\text{rang}('S') - \text{rang}('A') + 10$ (les dix compteurs des chiffres sont rangés devant), donc à l'indice 18 (S est la 19^{ième} lettre de l'alphabet) + 10, soit 28;
- si le caractère est une minuscule, le rang de cette minuscule diminué du rang du caractère 'a', exemple le caractère est 's', le compteur correspondant est à l'indice $\text{rang}('s') - \text{rang}('a') + 36$ (les dix compteurs des chiffres ainsi que les 26 compteurs des majuscules sont rangés devant), donc à l'indice 18 (s est la 19^{ième} lettre de l'alphabet) + 36, soit 54.



Rappel :

la fonction `rang()` existe pour les caractères, a le profil

fonction `rang(car : in caractere)` renvoie `entier_naturel`;

et renvoie le rang du caractère passé en paramètre. Les rangs des chiffres sont consécutifs et rangés dans l'ordre, ainsi que les rangs des majuscules et celui des minuscules.