

TP4 - COMPTAGE DE CARACTÈRES - ARITHMÉTIQUE SUR LES CODES ASCII

Le but de ce TP est de réaliser un programme permettant de lire puis d'analyser une suite de caractères saisie par l'utilisateur. Il permet notamment de montrer les opérations réalisables sur des caractères. Il permet aussi de voir l'encodage des caractères grâce aux codes ASCII (<http://en.wikipedia.org/wiki/ASCII>).

Analyse de caractères

Il s'agit ici de faire un programme qui permet de lire une suite de caractères au clavier puis de les analyser au fur à mesure. La suite est terminée lorsque l'utilisateur tape un point ('.').

Les caractères seront lus un par un en utilisant une seule variable.

L'analyse demandée consiste à compter, puis à afficher en fin de programme :

- le nombre d'occurrences des 10 chiffres (chiffres de '0' à '9')
- le nombre de caractères alphabétiques
- le nombre total de caractères d'espacement : espace (' '), tabulation ('\t') et retour à la ligne ('\n')

Préparation

1. Préparer des exemples en complétant les colonnes non grisées du tableau ci-dessous.

Valeurs fournies au clavier (Donner une liste d'au moins 10 caractères dans chacune des cases ci-dessous)	Valeur que devrait afficher le Programme	Résultat affiché après exécution	Le fonctionnement est-il conforme ?
Bonjour 1001 pattes et 106 voitures09.	5 espacements 23 caract alpha 10 chiffres ...		

2. Écrire en notation algorithmique chacun des sous-programmes suivants qui réalise un test sur un caractère et retourne un booléen :
 - estUnChiffre()
 - estUneLettre()
 - estUnEspacement()

3. Écrire le programme principal. Comme indiqué précédemment, chaque caractère doit être lu puis analysé par les sous-programmes dans le but d'incrémenter les compteurs.

Réalisation en séance

1. Créer un nouveau répertoire TLP_CPP\TP4 dans lequel on copie le fichier TLP_CPP\modele.cpp en le renommant cpt_char.cpp.
2. Ouvrir, dans **Code::Blocks**, un nouveau projet ayant pour fichier source cpt_char.cpp.
3. Traduire l'algorithme en C++ et l'éditer dans **Code::Blocks**.

La lecture des caractères doit se faire à l'aide de la syntaxe suivante pour comptabiliser les caractères d'espacement.

```
cin >> noskipws >> caract;
```

4. Tester le programme. Pour cela vous utiliserez les jeux de test préparés précédemment et vous remplirez le tableau (cases grisées) pour comparer les résultats attendus et ceux que vous avez.
5. Corriger le code si nécessaire puis conclure.

Code ASCII et transformation de caractères

Dans cette partie, il s'agit d'étudier l'encodage ASCII des caractères. Pour cela nous allons ajouter des fonctionnalités au programme de la partie précédente.

La première fonctionnalité consiste à afficher une variable de type caractère dans différents formats :

- Le caractère
- Code ASCII en hexadécimal (base 16)
- Code ASCII en décimal (base 10)

La bibliothèque `iostream` du langage C++ fournit des fonctions de formatage de l'affichage dans différentes bases. Par exemple, pour afficher un entier en hexadécimal puis en décimal on peut écrire :

```
cout << hex << "0x" << monEntier << " " << dec << monEntier;
```

La deuxième fonctionnalité consiste à transformer un caractère alphabétique en minuscule (s'il s'agit d'une majuscule), et inversement. On doit s'aider des codes ASCII fournis dans le tableau ci-dessous pour déduire comment tester si un caractère est en majuscule ou en minuscule.

Caractères	Codes en hexadécimal	Codes en décimal
'a' jusqu'à 'z'	0x61 jusqu'à 0x7a	97 jusqu'à 122
'A' jusqu'à 'Z'	0x41 jusqu'à 0x5a	65 jusqu'à 90

En reprenant le programme de la première partie du TP, on effectuera pour chaque caractère alphabétique la transformation des majuscules en minuscules et vice versa. Quel que soit le caractère, on fera un affichage de l'encodage : valeur hexadécimale et décimale du caractère.

Préparation

1. Préparer des exemples en complétant les colonnes non grisées du tableau ci-dessous.

Valeurs fournies au clavier	Valeur que devrait afficher le Programme	Résultat affiché après exécution	Le fonctionnement est-il conforme ?
C'est 1 Test.	c 0x63 99 ' 0x27 39 E 0x45 69 2 espacements 8 caract alpha ...		

2. Écrire en notation algorithmique chacun des sous-programmes suivants :
 - afficheCodeCaractere() : affiche dans les 3 formats un caractère. Pour afficher les valeurs hexadécimale et décimale du caractère, il faut affecter le caractère (ie son code ASCII) à une variable de type entier.
 - estUneMajuscule()
 - minus2majusc()
 - majusc2minusc()
3. Écrire une nouvelle version du programme principal précédent en intégrant les appels aux nouveaux sous-programmes.

Réalisation en séance

1. Copier le fichier TLP_CPP\TP4\cpt_char.cpp dans le dossier TLP_CPP\TP4 en le renommant transformation_char.cpp.
2. Ouvrir, dans **Code::Blocks**, un nouveau projet ainsi que le fichier transformation_char.cpp.
3. Traduire l'algorithme en C++ et l'éditer dans **Code::Blocks**.
4. Tester le programme. Pour cela vous utiliserez les exemples d'utilisation préparés précédemment et vous remplirez le tableau (cases grisées) pour comparer les résultats attendus et ceux que vous avez.
5. Corriger le code si nécessaire puis conclure.

Utilisation des bibliothèques standard

Dans cette partie nous allons reprendre les fonctions que vous avez créées dans les parties précédentes en utilisant des bibliothèques standard du langage C. Vous remplacerez les fonctions que vous avez développées avec les fonctions fournies par les bibliothèques.

Voici une liste de quelques fonctions disponibles : `isalpha()`, `isalnum()`, `isdigit()`, `isspace()`, `isupper()`, `islower()`, `tolower()`, `toupper()`, etc.

Ces fonctions sont documentées depuis la création des bibliothèques standard dans les pages de manuels.

- Accès en ligne : <http://man.he.net/>
- Accès à partir des fichiers `ctype-1.pdf` et `ctype-2.pdf` disponibles sur Moodle.

Préparation

1. Lire les pages de manuels et donner le prototype de la fonction `isdigit()`. Quel est le type du paramètre en entrée ? Quel est le type de la valeur retournée par la fonction ?
2. Quelle est la valeur retournée par la fonction en `isdigit()` en cas d'échec (la valeur donnée en entrée ne correspond pas à un chiffre) ?
3. Lire les pages de manuels et donner le prototype de la fonction `tolower()`. Quel est le type du paramètre en entrée ? Quel est le type de la valeur retournée par la fonction ?
4. Quelle est la valeur retournée par la fonction en `tolower()` en cas d'échec (la valeur donnée en entrée ne peut pas être transformée en minuscule) ?

Réalisation en séance

1. Copier le fichier `TLP_CPP\TP4\transformation_char.cpp` dans le dossier `TLP_CPP\TP4` en le renommant `transformlib_char.cpp`.
2. Ouvrir, dans **Code::Blocks**, un nouveau projet ainsi que le fichier `transformlib_char.cpp`.
3. Éditer le fichier source dans **Code::Blocks**. Remplacer tous les appels de sous-programmes de traitement des caractères par ceux fournis avec les bibliothèques standard.
4. Tester le programme. Pour cela vous utiliserez à nouveau les exemples d'utilisation préparés précédemment et vous remplirez le tableau (cases grisées) pour comparer les résultats attendus et ceux que vous avez.
5. Corriger le code si nécessaire puis conclure.