TP Codage des caractères

Compte-rendu informatique à rendre

Objectifs du TP :

- ***manipuler des variables de type char et des chaînes de caractères constantes en langage C dans différents environnements.***

Rappels :
- une variable du type char est occupe en mémoire du système **un seul octet**.
- une **expression entre apostrophes (ou quotes)** est une **constante de type char**. Exemple : 'a'. Donc une **expression entre apostrophes (ou quotes)** ne doit contenir qu'un seul caractère.
- une **expression entre guillemets** est une **chaîne de caractères constante**. Exemple : "toto". Elle peut contenir plusieurs caractères. Le cas échéant, elle doit donc être stockée dans **un tableau de caractères**.

Documents à rendre :

- 4 programmes sources (un par partie);
- Un compte rendu réalisé avec Libre Office Writer.

TRAVVAIL A REALISER :

1. Etude du type char en utilisant **CodeBlocks sur Windows**

Rappel : le préfixe 0x indique que le nombre est écrit en hexadécimal.

1.1. Déclarer une variable du type char, relever son adresse et le nombre d'octets qu'elle occupe.
1.2. **Compléter** le programme en affectant successivement les constantes du tableau ci-dessous dans votre variable du type char. Après chaque affectation, il faudra **observer le contenu de la case mémoire occupée** par la variable (utilisation de la fenêtre "Memory dump") et **afficher cette variable**, afin de **compléter le tableau**.

Valeur affectée	Valeur en mémoire	Valeur affichée
64		
43		
171		
'7'		
'F'		
0x65		
0x5E		
'é'		
0x82		

Remarques :
- Quand on affecte la constante caractère 'é' dans la variable de type char, on place dans l'octet correspondant le code 0xE9 car **CodeBlocks utilise la page Windows-1252**. Ensuite quand on affiche le caractère sur la console, le caractère affiché est le Ú car la **console utilise la page de code 850 définie par IBM** (cf. cours sur le codage des caractères).
- Pour les codes ASCII compris entre 0x00 et 0x7F, les caractères codés par les 2 pages sont identiques (respect du codage ASCII standard sur 7 bits). En revanche pour les codes compris entre 0x80 et 0xFF, les caractères codés diffèrent d'une page à l'autre. D'où des problèmes d'affichage sur la console de Windows pour ces derniers.

1.3. Justifier les valeurs en mémoire des 5 premières lignes du tableau.

1.4. Affecter une valeur dans la variable de type char afin d'afficher le caractère à.

2. Manipulation de chaînes de caractères constantes en utilisant **CodeBlocks sur Windows**

2.1. Créer un nouveau projet et modifier le fichier source "main.cpp" comme indiqué ci-dessous :

```
#include <iostream>
using namespace std;

int main()
{
	cout << "Bienvenue à Puteaux\n"; // Affichage d'une chaine de caractères constante
	return 0;
}
```

2.2. Générer un exécutable et lancer le. Quel est le problème et d'où vient-il ?

On peut obtenir le même résultat en plaçant la chaîne de caractères constante dans un tableau de caractères initialisé lors de sa déclaration, avant son affichage. Cf. code source ci-dessous :

```
 #include <iostream>
using namespace std;

int main()
{
	char tab [] = " Bienvenue à Puteaux\n"; // Déclaration d'un tableau de caractères de nom
                                                // tab initialisé lors de sa déclaration

	// utilisation de l'opérateur sizeof() afin de déterminer le nombre d'octets occupés par
	// la variable tab
	cout <<" Taille de l'espace memoire occupe par la variable tab : " << sizeof(tab) ;
	cout << endl;
	cout << tab;    // affiche la chaîne de caractères contenue dans la tableau tab
                        // et pas l'adresse de l'élément d'indice 0 du tableau tab.
// FONCTIONNE DE CETTE FAÇON QUE POUR LES TABLEAUX DE CARACTÈRES
	return 0;
}
```

2.3. Modifier votre programme avec le code source donné ci-dessus. Générer un exécutable et exécuter le en mode débugage afin de compléter le tableau suivant en indiquant les adresses et les contenus des 21 octets occupés par la variable tab dans l'espace d'adressage du programme (utilisation de la fenêtre "Memory dump").

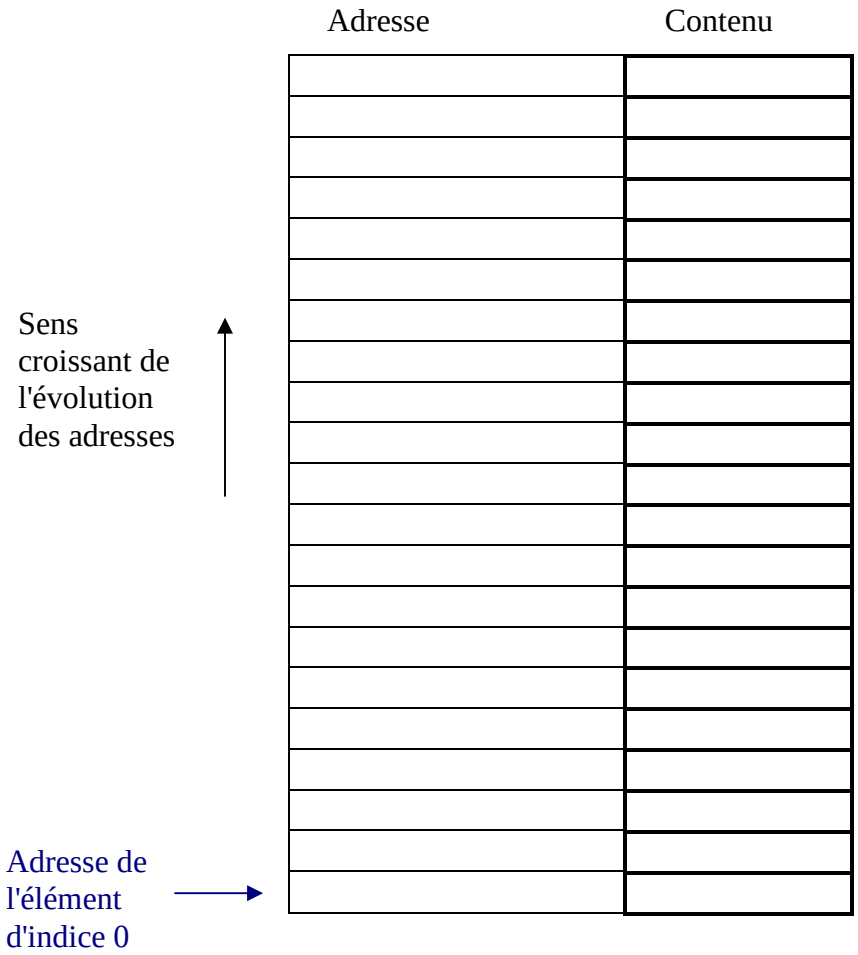


Figure 1 : Adresses et contenus des octets occupés par la variable tableau *tab*

2.4 Indiquer les codes ASCII des caractères suivants : 'B', 'a', ' ' (espace), 'x', '\n', 'à'. A quoi correspond le caractère constant '\n' ?

Remarque : une chaîne de caractères C est toujours terminée par le caractère nul de code ASCII 0x00.

2.5 La console utilise la page de code 850 définie par IBM. Quel est le code ASCII du caractère 'à' dans cette page de code? Convertir ce code ASCII en octal.

Afin d'afficher correctement le caractère 'à', il faut modifier l'initialisation du tableau tab et placer, dans la chaîne de caractères constante, le code ASCII sous forme octale ou hexadécimale du caractère 'à' en utilisant l'antislash.
= > 2 déclarations possibles du tableau tab :

```
char tab [] = " Bienvenue \x85 Puteaux\n";   // utilisation du code ASCII
                                             // du caractère 'à' sous forme hexadécimale
char tab [] = " Bienvenue \205 Puteaux\n";   // utilisation du code ASCII
                                             // du caractère 'à' sous forme octale
```

2.6 Vérifier que les 2 possibilités fonctionnent. Le nombre d'octets occupés par la variable tab est-il modifié?

2.7 Réaliser un programme qui affiche correctement le message suivant :

Bienvenue aux élèves d'IRIS

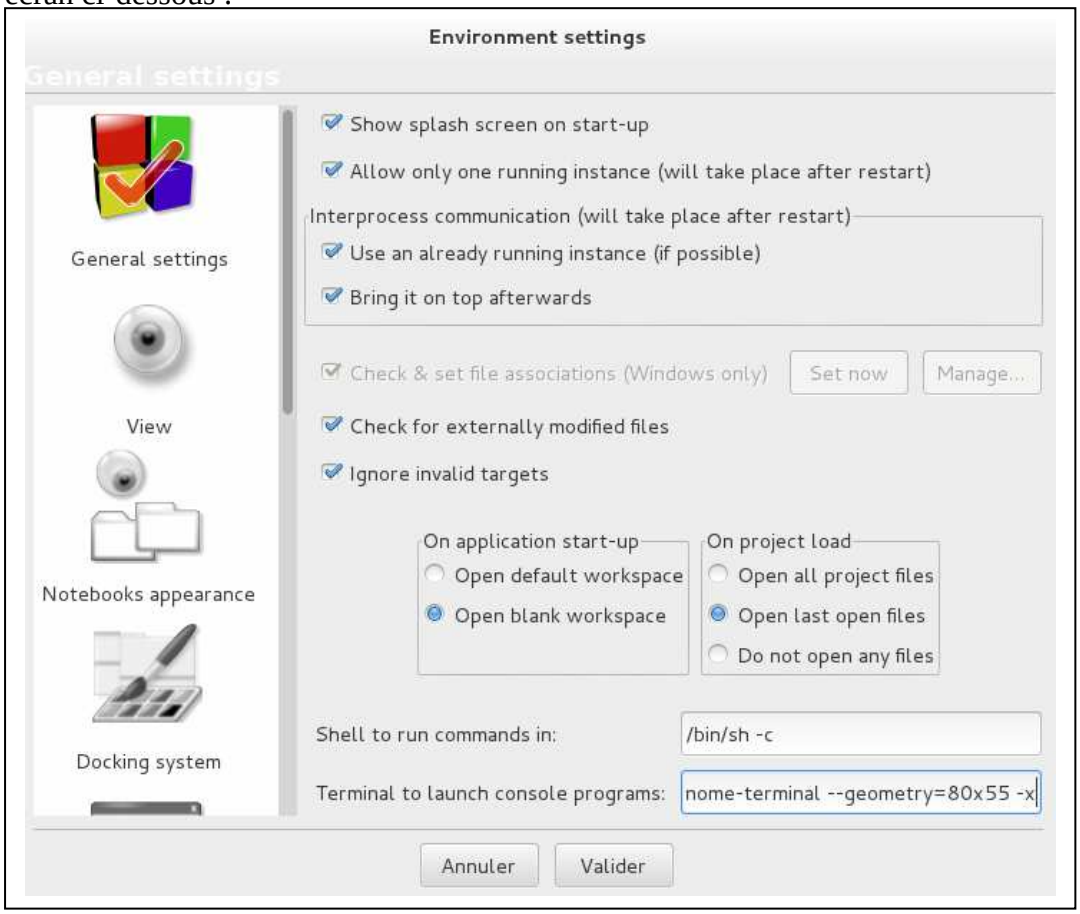
3. Etude du type char en utilisant **CodeBlocks sur Fedora**

Pour coder les caractères , **CodeBlocks et la console** utilisent **le code UTF-8.**

Dans le menu Settings/Environment, modifier le "terminal to launch console programs" afin d'utiliser l'émulateur de terminal "Gnome terminal" :

gnome-terminal --geometry=80x55 -x

Cf. copie d'écran ci-dessous :



3.1. Déclarer une variable du type char, relever son adresse et le nombre d'octets qu'elle occupe.
3.2. **Compléter** le programme en affectant successivement les constantes du tableau ci-dessous dans votre variable du type char. Après chaque affectation, il faudra **observer le contenu de la case mémoire occupée** par la variable (utilisation de la fenêtre "Memory dump") et **afficher cette variable**, afin de **compléter le tableau**.

Valeur affectée	Valeur en mémoire	Valeur affichée
64		
0x5E		
0x65		
171		
'é'		

3.3 Analyse des résultats obtenus à la question précédente :
- le code 0xAB = $171_{(10)}$ est-il un code UTF-8 valide ?
- quel est le code UTF-8 du caractère 'é'? Quel est le nombre d'octets de ce code? Est-il possible de stocker ce code dans une variable de type char ?

4. Manipulation de chaînes de caractères constantes en utilisant **CodeBlocks sur Fedora**

4.1. Créer un nouveau projet et modifier le fichier source "main.cpp" comme indiqué ci-dessous :

```
 #include <iostream>
using namespace std;

int main()
{
        char tab [] = " étoile\n";        // Déclaration d'un tableau de caractères de nom
                                          // tab initialisé lors de sa déclaration

        // utilisation de l'opérateur sizeof() afin de déterminer le nombre d'octets occupés par
        // la variable tab
        cout <<" Taille de l'espace memoire occupe par la variable tab : " << sizeof(tab) ;
        cout << endl;
        cout << tab;    // affiche la chaîne de caractères contenue dans la tableau tab
                        // et pas l'adresse de l'élément d'indice 0 du tableau tab.
// FONCTIONNE DE CETTE FAÇON QUE POUR LES TABLEAUX DE CARACTÈRES

        return 0;
}
```

4.2 Générer un exécutable et exécuter le en mode débugage afin de compléter le tableau suivant en indiquant les adresses et les contenus des 9 octets occupés par la variable tab dans l'espace d'adressage du programme (utilisation de la fenêtre "Memory dump").

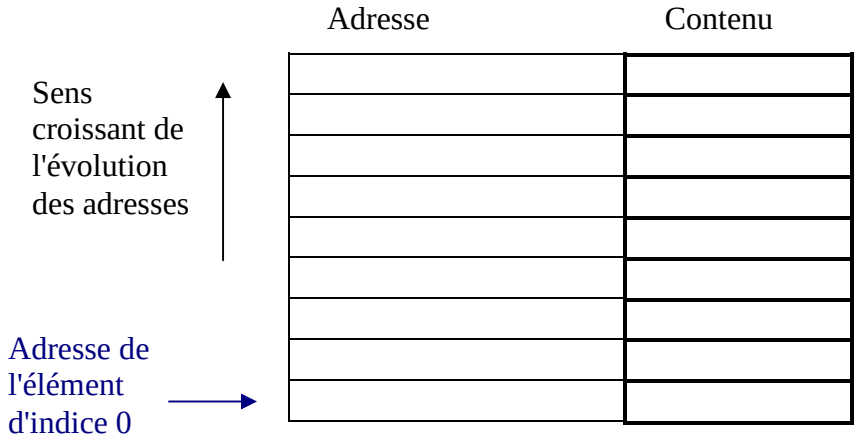


Figure 1 : Adresses et contenus des octets occupés par la variable tableau *tab*

4.3 Quel est le nombre d'octets occupés par le code UTF-8 du caractère 'é' ?

Remarque : la déclaration / initialisation suivante du tableau tab donne le même résultat.

```
char tab[] = "\xC3\xA9toile\n";
```

4.4 Vérifier que la remarque précédente est vraie.

4.5 Quel est le point de code du caractère è dans le répertoire Unicode ? (consulter le site http://www.unicode.org/fr/). En déduire le code UTF-8 de ce caractère.

4.6 Quels sont les points de code des caractères grecques Δ et θ dans le répertoire Unicode ? (consulter le site http://www.unicode.org/fr/). En déduire les codes UTF-8 de ces 2 caractères.

4.7 Réaliser un programme qui affiche correctement les messages suivants :

élèves
Δ et θ

4.8 Quelle est la taille **minimale** d'un tableau de caractères pouvant contenir la dernière chaîne de caractères constante? **Justifier et tester** votre réponse.