Introduction à la programmation
Travaux pratiques: séance 13
INFO0201-1

X. Baumans
(xavier.baumans@ulg.ac.be)

23/04/2014

Programme de la séance

- Chaînes de caractères
  $\rightarrow$ Gestion de texte avec un tableau de caractères (**char**)
  - Manipulations d'une suite de caractères contigus
  - Entrée-sortie de texte

Rappel : les caractères

- En C/C++, les caractères sont stockés dans des variables de type **char**. Ces variables contiennent en fait des nombres compris entre 0 et 127 qui représentent chacun un caractère différent.

Rappel : les caractères

- En C/C++, les caractères sont stockés dans des variables de type **char**. Ces variables contiennent en fait des nombres compris entre 0 et 127 qui représentent chacun un caractère différent.
- La correspondance entre un nombre et un caractère est établie dans une table appellée code **ASCII**.

Rappel : les caractères

- En C/C++, les caractères sont stockés dans des variables de type **char**. Ces variables contiennent en fait des nombres compris entre 0 et 127 qui représentent chacun un caractère différent.
- La correspondance entre un nombre et un caractère est établie dans une table appellée code **ASCII**.
- Le code correspondant à la lettre **A** est le nombre 65. On peut ainsi écrire indifféremment :

```
1 char une_lettre = 'A'; // --> le caractère A
2 une_lettre = 65; // --> aussi le caractère A
```

Pour indiquer qu'on considère un caractère comme sa valeur numérique correspondante dans le code ASCII, on entoure ledit caractère avec des **guillemets simples** : 'A', 'L' ou 'P' etc...

Rappel : les caractères - le code ASCII

Dec = Decimal; Hex = Hexadecimal; Char = Character

Dec	Hex	Char	Dec	Hex	Char	Dec	Hex	Char	Dec	Hex	Char
0	00	Null	32	20	Space	64	40	@	96	60	`
1	01	Start of heading	33	21	!	65	41	A	97	61	a
2	02	Start of text	34	22	"	66	42	B	98	62	b
3	03	End of text	35	23	#	67	43	C	99	63	c
4	04	End of transmit	36	24	$	68	44	D	100	64	d
5	05	Enquiry	37	25	%	69	45	E	101	65	e
6	06	Acknowledge	38	26	&	70	46	F	102	66	f
7	07	Audible bell	39	27	'	71	47	G	103	67	g
8	08	Backspace	40	28	(	72	48	H	104	68	h
9	09	Horizontal tab	41	29	)	73	49	I	105	69	i
10	0A	Line feed	42	2A	*	74	4A	J	106	6A	j
11	0B	Vertical tab	43	2B	+	75	4B	K	107	6B	k
12	0C	Form feed	44	2C	,	76	4C	L	108	6C	l
13	0D	Carriage return	45	2D	-	77	4D	M	109	6D	m
14	0E	Shift out	46	2E	.	78	4E	N	110	6E	n
15	0F	Shift in	47	2F	/	79	4F	O	111	6F	o
16	10	Data link escape	48	30	0	80	50	P	112	70	p
17	11	Device control 1	49	31	1	81	51	Q	113	71	q
18	12	Device control 2	50	32	2	82	52	R	114	72	r
19	13	Device control 3	51	33	3	83	53	S	115	73	s
20	14	Device control 4	52	34	4	84	54	T	116	74	t
21	15	Neg. acknowledge	53	35	5	85	55	U	117	75	u
22	16	Synchronous idle	54	36	6	86	56	V	118	76	v
23	17	End trans. block	55	37	7	87	57	W	119	77	w
24	18	Cancel	56	38	8	88	58	X	120	78	x
25	19	End of medium	57	39	9	89	59	Y	121	79	y
26	1A	Substitution	58	3A	:	90	5A	Z	122	7A	z
27	1B	Escape	59	3B	;	91	5B	[	123	7B	{
28	1C	File separator	60	3C	<	92	5C	\	124	7C	|
29	1D	Group separator	61	3D	=	93	5D	]	125	7D	}
30	1E	Record separator	62	3E	>	94	5E	^	126	7E	~
31	1F	Unit separator	63	3F	?	95	5F	_	127	7F	□

Rappel : les caractères

- Affichage des caractères : lors de l'affichage d'une variable de type **char** avec cout, le programme sait qu'il s'agit d'un caractère et affiche le caractère correspondant au code ASCII stocké dans la variable, au lieu du code ASCII lui-même.

```
1 char une_lettre = 'A'; // place 65 dans la variable
2 cout << une_lettre << endl; // affiche A
3 une_lettre = 66; // la lettre B en code ASCII
4 cout << une_lettre << endl; // affiche B
```

Chaînes de caractères

Chaîne de caractères

Une **chaîne de caractères** est une suite de caractères contigus stockés dans un tableau de type **char** et dont le dernier élément est le caractère `null '\0'`.

Pour gérer de manière pratique du texte (un ensemble de caractères), on utilise des tableaux de caractères, qu'on appelle des chaînes de caractères (*string* en anglais).

Pour indiquer que le texte est terminé, on place un élément spécial (caractère `null`) dans le dernier élément de la chaîne. Cela peut-être le dernier élément du tableau, mais pas nécessairement. En effet, la chaîne de caractère peut être plus courte que le tableau dans lequel elle est stockée.
Le nombre total de caractères dans une chaîne de caractères est donc égal à la taille de la chaîne de caractères + 1.

Déclarer et initialiser des chaînes de caractères

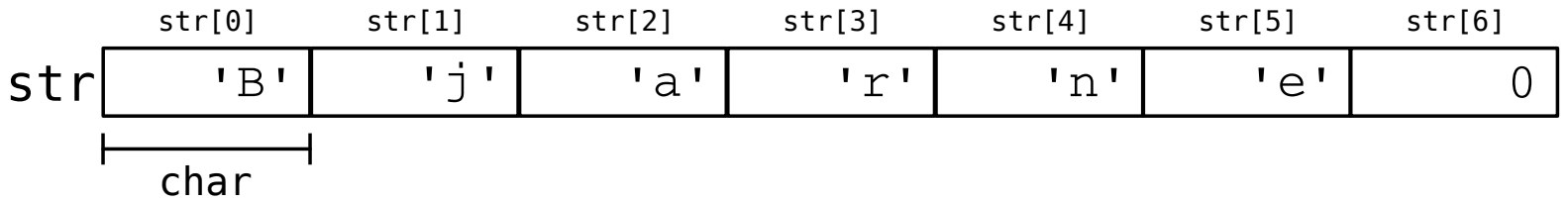
Pour déclarer et initialiser une chaîne de caractères, on déclare un tableau de type **char** et on l'initialise en écrivant la chaîne de caractères entre **guillemets doubles**.
Cela signale qu'il s'agit d'une chaîne de caractère et correspond à initialiser le tableau avec des chacun des caractères de la chaîne en terminant par le caractère **null** (code ASCII = 0)

```
1 char str[] = "Bjarne";
```

est équivalent à

```
1 char str[7] = {'B', 'j', 'a', 'r', 'n', 'e', '\0'};
```

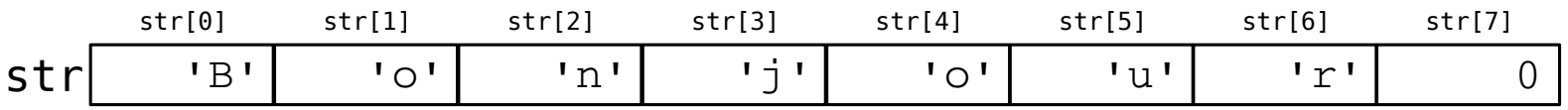
Représentation du tableau dans la mémoire :



Afficher des chaînes de caractères

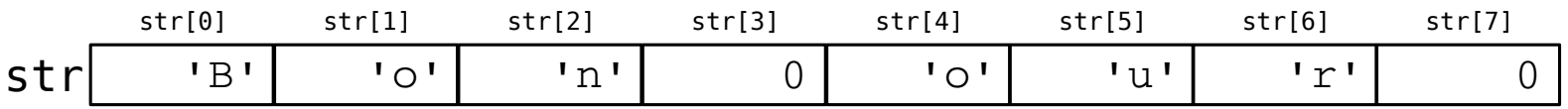
On ne peut normalement pas afficher un tableau entier de manière immédiate avec l'objet cout.
Les chaînes de caractères sont une exception : on peut afficher une chaîne de caractère en passant directement le nom du tableau qui la contient à l'objet cout :

```
1 char str[] = "Bonjour";
2 cout << str; // Affiche Bonjour
```



Le système affiche tous les caractères de la chaîne jusqu'à ce que le caractère **null** soit rencontré.

```
1 str[3] = 0;
2 cout << str; // Affiche Bon
```



Saisir des chaînes de caractères au clavier

L'objet cin permet lui aussi de lire directement des chaînes de caractères complètes. Pour cela, il suffit de signaler à cin qu'il doit placer la saisie clavier dans un tableau d'éléments de type char.

```
1 char buffer[28];
2 cin >> buffer; // ATTENTION: max 27 caractères !!!
```

Le système va copier tous les caractères saisis au clavier dans le tableau de caractères jusqu'à ce qu'un espace soit rencontré. Il placera alors automatiquement un caractère nul à la fin de la chaîne.

ATTENTION : le tableau doit être de taille suffisante pour pouvoir accueillir toute la chaîne de caractères (caractère null de fin y compris). Si ce n'est pas le cas, le système tentera d'écrire dans des éléments situés derrière le tableau déclaré, qui n'existent pas, et cela ne fonctionnera pas !

Chaînes de caractères dans des fichiers

- Écrire dans un fichier : l'écriture d'une chaîne de caractères dans un fichier fonctionne de la même manière qu'avec cout :

```
1 char buffer[28] = "Hello";
2 mon_fichier << buffer;
```

Chaînes de caractères dans des fichiers

- Écrire dans un fichier : l'écriture d'une chaîne de caractères dans un fichier fonctionne de la même manière qu'avec cout :

```
1 char buffer[28] = "Hello";
2 mon_fichier << buffer;
```

- Lire dans un fichier : la lecture d'une chaîne de caractères fonctionne de la même manière qu'avec cin :

```
1 char buffer[256];
2 mon_fichier >> buffer;
```

Comme avec cin, la lecture s'interrompt dès qu'un caractère 'espace' est rencontré.

Opérations sur les chaînes de caractères

Avec **tous** les tableaux, donc les chaînes de caractères aussi, il est impossible de copier directement un tableau dans l'autre :

```
1 char buffer[256] = "Bonjour";
2 char buffer2[256] = "Hello";
3 buffer_2 = buffer; // NON !!!
```

Il n'est pas permis de copier un tableau dans un autre en utilisant l'opérateur d'affectation (explication à la prochaine séance).
Il faut donc réaliser les opérations "à la main", caractère par caractère.

Relations utiles avec les chaînes de caractères

Tous les caractères correspondant à un nombre précis, il existe des relations utiles entre certains caractères.

- Majuscule-Minuscule : les lettres A-Z et a-z se suivent dans l'ordre alphabétique dans le code ASCII. La lettre A possède le code 65, a $= 97$, B $= 66$, b $= 98$, etc...
  Il est donc possible de convertir une lettre en la majuscule ou la minuscule correspondante en soustrayant ou ajoutant $97 - 65 = 32$ à la valeur de la lettre en question.

Relations utiles avec les chaînes de caractères

Tous les caractères correspondant à un nombre précis, il existe des relations utiles entre certains caractères.

- Majuscule-Minuscule : les lettres A-Z et a-z se suivent dans l'ordre alphabétique dans le code ASCII. La lettre A possède le code 65, a $= 97$, B $= 66$, b $= 98$, etc...
  Il est donc possible de convertir une lettre en la majuscule ou la minuscule correspondante en soustrayant ou ajoutant $97 - 65 = 32$ à la valeur de la lettre en question.
- Valeur numérique d'un chiffre : de la même manière, les chiffres $0 - 9$ se suivent dans le code ASCII. Puisque le "0" a le code 48, il suffit de soustraire 48 à la valeur d'un chiffre codé en ASCII pour obtenir la valeur numérique correspondante.

Relations utiles avec les chaînes de caractères

Tous les caractères correspondant à un nombre précis, il existe des relations utiles entre certains caractères.

- Majuscule-Minuscule : les lettres A-Z et a-z se suivent dans l'ordre alphabétique dans le code ASCII. La lettre A possède le code 65, a = 97, B = 66, b = 98, etc...
  Il est donc possible de convertir une lettre en la majuscule ou la minuscule correspondante en soustrayant ou ajoutant $97 - 65 = 32$ à la valeur de la lettre en question.
- Valeur numérique d'un chiffre : de la même manière, les chiffres $0 - 9$ se suivent dans le code ASCII. Puisque le "0" a le code 48, il suffit de soustraire 48 à la valeur d'un chiffre codé en ASCII pour obtenir la valeur numérique correspondante.
- Caractères non imprimables : les caractères dont le code ASCII est compris entre 0 et 31 sont des caractères dits "non-imprimables" qu'il n'est donc pas possible d'afficher à l'écran.

Fonctions de caractères standards

Il existe une série de fonctions de la bibliothèque standard qui permettent d'effectuer des opérations sur les chaînes de caractères. Pour les utiliser, il faut inclure la librairie **cstring**

```
1 #include <cstring>
```

Toute la documentation contenant la liste des fonctions contenues dans la librairie ainsi que leurs paramètres et la manière de les utiliser est disponible à l'adresse
`http://www.cplusplus.com/reference/cstring/`

Fonctions de caractères standards

Quelques fonctions de chaînes de caractères standard :

- strlen(char[] str) : retourne la longueur de la chaîne de caractères ;

Fonctions de caractères standards

Quelques fonctions de chaînes de caractères standard :

- strlen(char[] str) : retourne la longueur de la chaîne de caractères ;
- strcpy(char[] destination, char[] source) : copie la chaîne *source* dans la chaîne *destination* ;

Fonctions de caractères standards

Quelques fonctions de chaînes de caractères standard :

- strlen(char[] str) : retourne la longueur de la chaîne de caractères ;
- strcpy(char[] destination, char[] source) : copie la chaîne *source* dans la chaîne *destination* ;
- strcat(char[] destination, char[] source) : ajoute une copie de *source* à la suite de la chaîne *destination* ;

Fonctions de caractères standards

Quelques fonctions de chaînes de caractères standard :

- strlen(char[] str) : retourne la longueur de la chaîne de caractères ;
- strcpy(char[] destination, char[] source) : copie la chaîne *source* dans la chaîne *destination* ;
- strcat(char[] destination, char[] source) : ajoute une copie de *source* à la suite de la chaîne *destination* ;
- strcmp(char[] str1, char[] str2) : compare les chaînes *str1* et *str2* et renvoit :
  - 0 : si les chaînes sont identiques ;
  - > 0 : si le premier caractère différent a une valeur plus grande dans *str1* que dans *str2* ;
  - < 0 : dans le cas contraire.

Exercices (1/2)

Attention, la plupart des fonctions à concevoir pour ces exercices seront utiles pour les exercices ultérieurs.

1. Manipulations de chaînes de caractères
   - Ecrire une fonction qui prend en paramètres une chaîne de caractères et un caractère. La fonction retournera un entier correspondant au nombre de fois que le caractère apparaît dans la chaîne.
   - Ecrire une fonction qui prend en paramètre une chaîne de caractères et un booléen.
     - Si le booléen est **vrai** : modifier la chaîne de manière à ce que le premier caractère soit en majuscule ;
     - Si le booléen est **faux** : modifier la chaîne de manière à ce que le premier caractère soit en minuscule ;
   - Tester ces deux fonctions avec la chaîne de caractères “Prediction is very difficult, especially of the future. - Niels Bohr”.
   - Lire au clavier un certain nombre de prénoms (saisis avec ou sans majuscules) et les écrire ligne par ligne dans un fichier, tous avec la première lettre en majuscule et la longueur de leur prénom entre parenthèses.

Exercices (2/2)

2. Code de César - cryptage de texte.
   - Écrire une fonction de cryptage qui prend en paramètre une chaîne de caractère et une clé (=un nombre entier). La fonction va crypter la chaîne de caractères en décalant tous les caractères de la chaîne d'une valeur égale à la clé (exemple : la clé = 1, alors A $\rightarrow$ B, B $\rightarrow$ C, etc.)
     ATTENTION : veiller à ce que le décalage ait pour résultat un caractère dont la valeur est comprise entre 32 et 127 (caractères imprimables) en utilisant l'opérateur modulo.
   - Charger ligne par ligne un texte contenu dans le fichier message.txt disponible à l'adresse `http://spin02.phys.ulg.ac.be/` et place toutes les lignes à la suite l'une de l'autre dans une chaîne de caractères (taille max 1000 caractères) et afficher le texte à l'écran ;
   - Demander un mot de passe au clavier, sommer la valeur de tous les caractères du mot de passe pour obtenir la clé et crypter le texte avec cette clé ;
   - Afficher le texte crypté à l'écran et l'enregistrer dans un fichier appelé message_code.txt.