Ludivine LASSERRE

TP4 C++ - CHAINES DE CARACTERES

COMPTAGE DE CARACTERES

OBJECTIF :

Réaliser un programme qui permet de lire une suite de caractères au clavier puis de les analyser au fur et 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 (i.e. le nombre de 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')

REALISATION

1ERE ETAPE : INITIALISATION

La 1ère étape consistait à demander à un utilisateur d'entrer une suite de caractères au clavier jusqu'à ce qu'il entre le caractère '.'. Et de stocker ce résultat dans une variable. Pour ce faire j'ai utilisé les fonctions cout et cin (avec utilisation de noskipws qui permet de lire les caractères comme l'espace, la tabulation…).

```
char phrase[20];
char a;
int i = 0;
cout<<"Veuillez entrer votre texte : \n>>>"<<endl;
    do
    {
        cin>> noskipws >>a;
        phrase[i]=a;
        i++;
    } while(a!='.');
```

2EME ETAPE : RECUPERATION DES INFORMATIONS SUR LA CHAINE DE CARACTERES

Dans cette seconde étape, nous avions comme objectif de récupérer le nombre de lettres, chiffres, espaces de la chaîne de caractère sans utiliser les fonctions de la bibliothèque « cctype ».

J'ai donc codé les fonctions estUnChiffre(), estUneLettre(), estUnEspacement() qui prennent en entrée un caractères et un booléen et qui modifie la valeur de ce booléen.

```
void estUnChiffre(char a, bool &chiffre)
{
    chiffre = false;
    if((a>='0' && a<='9'))
        chiffre = true;
}


void estUneLettre(char a, bool &lettre)
{
    lettre = false;
    if((a>='a' && a<='z') || (a>='A' && a<='Z'))
        lettre = true;
```

```
}


void estUnEspacement(char a, bool &espacement)
{
    espacement = false;
    if(a==' ')
        espacement = true;
}


void estUneTabulation(char a, bool &tabulation)
{   tabulation = false;
    if(a=='\t')
        tabulation = true;
}
```

Après appels des fonctions, j'affiche les résultats grâce à cin et cout.

```
    cout<<"Il y a "<<nbLettre<<" lettres"<<endl;
    cout<<"Il y a "<<nbChiffre<<" chiffres"<<endl;
    cout<<"Il y a "<<nbEspacement<<" espacements"<<endl;
    cout<<"Il y a "<<nbTab<<" tabulations"<<endl;
```

RESULTATS

Dans cette partie, nous allons tester notre programme avec des valeurs concrètes. Pour vérifier notre résultat, nous allons les comparer avec les valeurs que nous devons obtenir combinées aux valeurs affichées en utilisant les fonctions de la bibliothèque cctype à savoir : isalpha(), isdigit(), isspace(). Ces fonctions vérifient respectivement si un caractère est alpha, si c'est un chiffre ou si c'est un espace.

Affichons nos résultats sous forme de tableau :

Valeurs fournies au clavier	**Valeur que devrait afficher le Programme**	**Résultat affiché après exécution**	**Résultat affiché avec utilisation des fonctions de la bibliothèque**	**Le fonctionnement est-il conforme ?**
Bonjour 1001 pattes et 106 voitures09.	5 espacements 23 caractères alpha 9 chiffres	5 espacements 23 caractères alpha 9 chiffres	5 espacements 23 caractères alpha 9 chiffres	OUI
Ceci est un test 1001	13 lettres 4 chiffres 4 espacements	13 lettres 4 chiffres 4 espacements	13 lettres 4 chiffres 4 espacements	OUI

On remarque que le fonctionnement de notre programme est conforme car nos résultats attendus sont les mêmes que les résultats obtenus.

CODE ASCII ET TRANSFORMATION DE CARACTERES

OBJECTIF

Dans cette partie, on étudie les codes ASCII des caractères en ajoutant deux fonctionnalités au programme de la partie précédente. La première fonctionnalité consiste à afficher une variable de type caractère dans différents formats :

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

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.

REALISATION

PREMIERE FONCTIONNALITE :

Pour la première fonctionnalité, j'ai créée une fonction afficheCodeCaractere() qui prend en paramètres d'entrée un caractère et qui l'affiche ainsi que son code hexadécimal puis décimal. J'utilise les fonctions hex et dec sur un caractère casté en tant qu'entier.

```
void afficheCodeCaractere(char a)
{
    cout<<a<<" "<<hex<<"0x"<<int(a)<<" "<<dec<<int(a)<<endl;
}
```

DEUXIEME FONCTIONNALITE :

Pour cette deuxième fonctionnalité, j'ai codé trois fonctions différentes :

- Fonction estUneMajuscule() :

Cette fonction vérifie pour un caractère donné si c'est une majuscule ou non en utilisant l'écriture décimale du nombre. Rappelons que :

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

```
void estUneMajuscule(char a, bool &maj)
{
    maj = false;
    if(int(a)>=65 && int(a)<=90)
        maj = true;
}
```

- Fonctions minus2majusc() et majusc2minus :

Pour ces fonctions j'ai utilisé le code ASCII du caractère. Ce code nous montre que pour passer de la minuscule à la majuscule il suffit de retirer 32 ou de rajouter 32 pour transformer une majuscule en minuscule.

```
void minus2majusc(char &a)
{
    a = int(a) - 32;
}

void majusc2minus(char &a)
{
    a = int(a) + 32;
}
```

RESULTATS

Dans cette partie je vais procéder à l'affichage des caractères de la phrase en hexadécimal et en décimal et la comparer grâce à la table ASCII.

PREMIERE FONCTIONNALITE :

Résultats du programme :

```
B 0x42 66
o 0x6f 111
n 0x6e 110
j 0x6a 106
o 0x6f 111
u 0x75 117
r 0x72 114
  0x20 32
1 0x31 49
0 0x30 48
0 0x30 48
1 0x31 49
```

ASCII TABLE

Decimal	Hex	Char	Decimal	Hex	Char	Decimal	Hex	Char	Decimal	Hex	Char
0	0	[NULL]	32	20	[SPACE]	64	40	@	96	60	`
1	1	[START OF HEADING]	33	21	!	65	41	A	97	61	a
2	2	[START OF TEXT]	34	22	"	66	42	B	98	62	b
3	3	[END OF TEXT]	35	23	#	67	43	C	99	63	c
4	4	[END OF TRANSMISSION]	36	24	$	68	44	D	100	64	d
5	5	[ENQUIRY]	37	25	%	69	45	E	101	65	e
6	6	[ACKNOWLEDGE]	38	26	&	70	46	F	102	66	f
7	7	[BELL]	39	27	'	71	47	G	103	67	g
8	8	[BACKSPACE]	40	28	(	72	48	H	104	68	h
9	9	[HORIZONTAL TAB]	41	29	)	73	49	I	105	69	i
10	A	[LINE FEED]	42	2A	*	74	4A	J	106	6A	j
11	B	[VERTICAL TAB]	43	2B	+	75	4B	K	107	6B	k
12	C	[FORM FEED]	44	2C	,	76	4C	L	108	6C	l
13	D	[CARRIAGE RETURN]	45	2D	-	77	4D	M	109	6D	m
14	E	[SHIFT OUT]	46	2E	.	78	4E	N	110	6E	n
15	F	[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	[NEGATIVE 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	[ENG OF 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	[SUBSTITUTE]	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	[DEL]

DEUXIEME FONCTIONNALITE :

Petit rappel : la deuxième fonctionnalité consiste à transformer un caractère alphabétique en minuscule (s'il s'agit d'une majuscule), et inversement. Dans cette partie nous allons comparer nos résultats obtenus avec nos propres fonctions avec les résultats obtenus en utilisant les fonctions de la bibliothèque cctype à savoir : islower(), isupper(), tolower(), toupper().

Présentons nos résultats sous forme de tableau :

Valeurs fournies au clavier	**Valeur que devrait afficher le Programme**	**Résultat affiché après exécution**	**Résultat affiché avec utilisation des fonctions de la bibliothèque**	**Le fonctionnement est-il conforme ?**
Bonjour 1001 pattes et 106 voitures09.	bONJOUR 1001 PATTES ET 106 VOITURES09	bONJOUR 1001 PATTES ET 106 VOITURES09	bONJOUR 1001 PATTES ET 106 VOITURES09	OUI
Ceci est un test 1001	cECI EST UN TEST 1001	cECI EST UN TEST 1001	cECI EST UN TEST 1001	OUI

On constate que le fonctionnement de notre programme est conforme car nos résultats attendus sont les mêmes que les résultats obtenus.

CONCLUSION

A travers ce TP nous avons appris à manipuler les chaînes de caractères. Nous avons vu comment compter les différents caractères d'une chaîne en la parcourant, comment transformer cette chaîne et nous avons également appris quelques spécificités du code ASCII des caractères.