

## TP N°6 :Chaînes de caractères en C

### Objectifs :

- Notions de base pour la gestion des chaînes de caractères en C.
- Prise en charge des chaînes de caractères avec des blancs.

### Énoncé :

Le C ne possède pas de type prédéfini chaîne de caractères, mais il propose de les représenter par des tableaux de caractères terminés par le caractère de code ASCII nul, et de les gérer par des fonctions prédéfinies en utilisant <string.h>.

### 1. Manipulations basiques

1.1 Dérouler manuellement le programme suivant puis l'exécuter. Comparer vos résultats manuels avec ceux de la machine.

```
1  #include <stdio.h>
2  #include <string.h>
3  #define TailleMax 20
4
5  int main()
6  {
7      char ch1[2*TailleMax+1] = "",  

8          ch2[TailleMax+1] = "Bonjour",  

9          ch3[TailleMax+1] = "Bonsoir",  

10         ch4[] = {'B', 'o', 'n', ' ', 'j', 'o', 'u', 'r', '\0'};
11
12     printf("\n Vous avez 4 chaines: \n la premiere, chaine 1, contient %s "  

13           "\n la deuxieme, chaine 2, contient %s"  

14           "\n la troisieme, chaine 3, contient %s"  

15           "\n enfin la quatrieme, chaine 4, contient %s.", ch1, ch2, ch3, ch4);
16     printf("\n 1. Comparaison de chaine 2 a chaine 4 :%d", strcmp(ch2, ch4));
17     printf("\n 2. Comparaison de chaine 2 a chaine 3 :%d", strcmp(ch1, ch2));
18     printf("\n 3. Comparaison de chaine 3 a chaine 2 :%d", strcmp(ch2, ch1));
19     printf("\n 4. Copions chaine 2 dans chaine 1");
20     strcpy(ch1, ch2);
21     printf("\n\nLa nouvelle valeur de chaine 1 est:%s", ch1);
22     printf("\n 5. Réalisons une concatenation de chaine 1 avec chaine 3 dans la premiere :");
23     strcat(ch1, ch3);
24     printf("\n\nLa nouvelle valeur de chaine 1 est:%s", ch1);
25
26     printf("\n 6. La longueur de chaine 1 est :%u", strlen(ch1));
27
28     printf("\n 7. Expliquer ce qui suit:");
29
30     ch1[10] = '\0';
31     printf("\n\n\t Longueur de chaine 1 devient :%u", strlen(ch1));
32     printf("\n\n\t et chaine 1 est maintenant :%s", ch1);
33
34     return 0;
35 }
```

1.2 Du point de vue du programmeur, Quelle est la différence entre ch2 et ch4 ?

### 2. Lecture de chaînes de caractère avec des blancs

2.1 Exécuter le programme suivant et commenter ce qu'il fait :

```

1 #include <stdio.h>
2 #include <stdlib.h>
3 #define MAX_CHAINE 30
4
5 char myChaine1[MAX_CHAINE+1],myChaine2[MAX_CHAINE+1],myChaine3[MAX_CHAINE+1];
6 int i;
7 int main()
8 {
9     printf("\n 1. Reecrire le texte suivant : Je suis algerien \n");
10    scanf("%s",myChaine1);
11    printf("\n 2. l'execution est arrivee a la ligne 11, c'est normal");
12    scanf("%s",myChaine2);
13    printf("\n 3. l'execution est arrivee a la ligne 13, c'est pas normal, il y a un scanf avant moi!");
14    scanf("%s",myChaine3);
15    printf("\n 4. l'execution est arrivee a la ligne 15! c'est pas normal, il y a des scanf avant moi!"
16        "\n 5. Essayons de comprendre ce qui c'est passe :");
17
18    printf("\n La chaine 1 contient : %s \n La chaine 2 contient : %s"
19        "\n La chaine 3 contient : %s",myChaine1,myChaine2,myChaine3);
20
21 }

```

2.2 Exécuter le programme suivant puis commenter ce qu'il fait :

```

1 #include <stdio.h>
2 #include <stdlib.h>
3 #define MAX_CHAINE 30
4
5 char myChaine[MAX_CHAINE+1];
6
7 int main()
8 {
9     printf("\n 1. Reecrire le texte suivant : Je suis algerien \n");
10    fgets(myChaine,MAX_CHAINE+1,stdin);
11
12    printf("\n La chaine contient : %s ",myChaine);
13
14 }

```

2.3 fgets joue-t-elle bien le jeu ? En vérité presque. Exécuter le code suivant puis commenter (Utiliser la table ASCII).

```

1 #include <stdio.h>
2 #include <stdlib.h>
3 #include <string.h>
4 #define MAX_CHAINE 20
5
6 char myChaine2[MAX_CHAINE+1],myChaine1[MAX_CHAINE+1]="Je suis algerien";
7 int i;
8
9 int main()
10 {
11     printf("\n 1. Une premiere chaine est initialisee a: %s ",myChaine1);
12     printf("\n 2. Reecrire le meme texte, nous allons le mettre dans une 2ieme chaine. \n");
13     fgets(myChaine2,MAX_CHAINE+1,stdin);
14
15     printf("\n 3. Comparons les deux chaines :");
16     if(strcmp(myChaine1,myChaine2)!=0)
17         printf("\n pour la machine les deux chaines sont differentes");
18
19     printf("\n 4. Essayons de voir plus clair, balisons les deux chaines."
20           "\n\tLa chaine 1 contient :-->%s<--\n\tLa chaine 2 contient :-->%s<--"
21           "\n Qu'est ce que vous remarquez?",myChaine1,myChaine2);
22     printf("\n\n 5. Essayons de voir de plus en plus clair:");
23     printf("\n La chaine 1 contient les caracteres de code ASCII:");
24     for(i=0;myChaine1[i];i++)printf("%d",myChaine1[i]);
25     printf("\n La chaine 2 contient les caracteres de code ASCII:");
26     for(i=0;myChaine2[i];i++)printf("%d",myChaine2[i]);
27
28
29 }

```

```

x - fget2

1. Une premiere chaine est initialisee a; Je suis algerien
2. Reecrire le meme texte, nous allons le mettre dans une 2ieme chaine.
Je suis algerien

3. Comparons les deux chaines :
pour la machine les deux chaines sont differentes
4. Essayons de voir plus clair, balisons les deux chaines.
   La chaine 1 contient :-->Je suis algerien<--
   La chaine 2 contient :-->Je suis algerien
<--
Qu'est ce que vous remarquez?

5. Essayons de voir de plus en plus clair:
La chaine 1 contient les caracteres de code ASCII: 74 101 32 115 117 105 115
32 97 108 103 101 114 105 101 110
La chaine 2 contient les caracteres de code ASCII: 74 101 32 115 117 105 115
32 97 108 103 101 114 105 101 110 10
Process returned 0 (0x0) execution time : 15.365 s
Press ENTER to continue.

```

2.4 Quel est le caractère de code ASCII 10 ?

2.5 Que vaut myChaine2[strlen(myChaine2)-1] ?

2.6 En se basant sur fgets (et la question 2.5), proposer une solution qui joue bien le jeu.

## Table ASCII

Dec Char	Dec Chr	Dec Char	Dec Chr
0 NUL (null)	32 Space	64 0	96 `
1 SOH (start of heading)	33 !	65 A	97 a
2 STX (start of text)	34 "	66 B	98 b
3 ETX (end of text)	35 #	67 C	99 c
4 EOT (end of transmission)	36 \$	68 D	100 d
5 ENQ (enquiry)	37 %	69 E	101 e
6 ACK (acknowledge)	38 &	70 F	102 f
7 BEL (bell)	39 ^	71 G	103 g
8 BS (backspace)	40 (	72 H	104 h
9 TAB (horizontal tab)	41 )	73 I	105 i
10 LF (NL line feed, new line)	42 *	74 J	106 j
11 VT (vertical tab)	43 +	75 K	107 k
12 FF (NP form feed, new page)	44 ,	76 L	108 l
13 CR (carriage return)	45 -	77 M	109 m
14 SO (shift out)	46 .	78 N	110 n
15 SI (shift in)	47 /	79 O	111 o
16 DLE (data link escape)	48 0	80 P	112 p
17 DC1 (device control 1)	49 1	81 Q	113 q
18 DC2 (device control 2)	50 2	82 R	114 r
19 DC3 (device control 3)	51 3	83 S	115 s
20 DC4 (device control 4)	52 4	84 T	116 t
21 NAK (negative acknowledge)	53 5	85 U	117 u
22 SYN (synchronous idle)	54 6	86 V	118 v
23 ETB (end of trans. block)	55 7	87 W	119 w
24 CAN (cancel)	56 8	88 X	120 x
25 EM (end of medium)	57 9	89 Y	121 y
26 SUB (substitute)	58 :	90 Z	122 z
27 ESC (escape)	59 ;	91 [	123 {
28 FS (file separator)	60 <	92 \	124
29 GS (group separator)	61 =	93 ]	125 }
30 RS (record separator)	62 >	94 ^	126 ~
31 US (unit separator)	63 ?	95 _	127 DEL