

# **Mise à niveau en C**

Tableaux et boucles

Enseignant: P. Bertin-Johannet

# Les tableaux

- Un tableau est une liste de taille finie d'éléments du même type
- Pour déclarer un tableau on précise:
  - Le type des éléments qu'il contient
  - Le nom de la variable
  - Le nombre d'éléments
- Par exemple pour un tableau de 5 entiers

```
int nom_tableau[5];
```

- On accède ensuite à l'élément n d'un tableau ainsi :

```
nom_tableau[n - 1]
```

# Exemple

Programme:

```
char tab [4];  
tab[0] = 1;  
tab[1] = 2;  
tab[2] = tab[0] + 2;
```

Mémoire:

Valeurs	Octets
	Octet 1
	Octet 2
	Octet 3
	Octet 4
	Octet 5
	Octet 6
	Octet 7
	Octet 8
	Octet 9

# Exemple

Programme:

```
char tab [4];
```

```
tab[0] = 1;
```

```
tab[1] = 2;
```

```
tab[2] = tab[0] + 2;
```

Mémoire:

Valeurs	Octets
	Octet 1
	Octet 2
	Octet 3
	Octet 4
	Octet 5
	Octet 6
	Octet 7
	Octet 8
	Octet 9

# Exemple

Programme:

```
char tab [4];
```

```
tab[0] = 1;
```

```
tab[1] = 2;
```

```
tab[2] = tab[0] + 2;
```

Mémoire:

Valeurs	Octets
1	Octet 1
	Octet 2
	Octet 3
	Octet 4
	Octet 5
	Octet 6
	Octet 7
	Octet 8
	Octet 9

# Exemple

Programme:

```
char tab [4];
```

```
tab[0] = 1;
```

```
tab[1] = 2;
```

```
tab[2] = tab[0] + 2;
```

Mémoire:

Valeurs	Octets
1	Octet 1
2	Octet 2
	Octet 3
	Octet 4
	Octet 5
	Octet 6
	Octet 7
	Octet 8
	Octet 9

# Exemple

Programme:

```
char tab [4];
```

```
tab[0] = 1;
```

```
tab[1] = 2;
```

```
tab[2] = tab[0] + 2;
```

Mémoire:

Valeurs	Octets
1	Octet 1
2	Octet 2
3	Octet 3
	Octet 4
	Octet 5
	Octet 6
	Octet 7
	Octet 8
	Octet 9

# Les chaines de caractères

- En C, une chaine de caractère est un tableau de `char` se terminant par la valeur zero

## Attention

Il ne s'agit pas du caractère '0' (qui vaut en ASCII 48) mais de '\0' (qui est en ASCII 0).

- Il faut donc toujours reserver un octet de plus pour une chaines de caractères.
- Le langage C permet d'écrire une chaine de caractères entre guillemets :  
`"Bonjour"`



# Exemple

Programme:

```
char tab [4] = "oui";  
tab[0]++;
```

Mémoire:

Valeurs	Octets
	Octet 1
	Octet 2
	Octet 3
	Octet 4
	Octet 5
	Octet 6
	Octet 7
	Octet 8
	Octet 9

# Exemple

Programme:

```
char tab [4] = "oui";
```

```
tab[0]++;
```

Mémoire:

Valeurs	Octets
111 (o)	Octet 1
117 (u)	Octet 2
105 (i)	Octet 3
0	Octet 4
	Octet 5
	Octet 6
	Octet 7
	Octet 8
	Octet 9

# Exemple

Programme:

```
char tab [4] = "oui";
```

```
tab[0]++;
```

Mémoire:

Valeurs	Octets
112 (p)	Octet 1
117 (u)	Octet 2
105 (i)	Octet 3
0	Octet 4
	Octet 5
	Octet 6
	Octet 7
	Octet 8
	Octet 9

# Les chaines de caractères

- On peut utiliser printf et scanf avec les chaines de caractères
- Le format à utiliser est %s



## Attention

Pour scanner une chaine de caractères avec scanf, il ne faut pas utiliser de &

# Example

Programme:

```
char tab [4];
```

```
scanf("%s", tab);
```

```
tab[0] -= 32;
```

```
printf("%s\n", tab);
```

# Exemple

Programme:

```
char tab [4];
```

```
scanf("%s", tab);
```

```
tab[0] -= 32;
```

```
printf("%s\n", tab);
```

Entrée utilisateur: > oui

# Exemple

Programme:

```
char tab [4];  
  
scanf("%s", tab);  
tab[0] -= 32;  
printf("%s\n", tab);
```

Entrée utilisateur: > oui

Sortie: > Oui

# La boucle while

- L’instruction `while` (en anglais : “tant que”) permet d’exécuter du code tant qu’une condition est vraie.

```
while(condition){  
    code  
}
```

- Les lignes de code entre les accolades seront répétées tant que la condition est vraie



# Exemple

Programme:

```
char tab [4] = "oui";  
int i = 0;  
while(tab[i] != 0){  
    tab[i]++;  
    i++;  
}
```

Mémoire:

Valeurs	Octets
	Octet 1
	Octet 2
	Octet 3
	Octet 4
	Octet 5
	Octet 6
	Octet 7

# Exemple

Programme:

```
char tab [4] = "oui";  
  
int i = 0;  
while(tab[i] != 0){  
    tab[i]++;  
    i++;  
}
```

Mémoire:

Valeurs	Octets
111 (o)	Octet 1
117 (u)	Octet 2
105 (i)	Octet 3
0	Octet 4
	Octet 5
	Octet 6
	Octet 7

# Exemple

Programme:

```
char tab [4] = "oui";  
int i = 0;  
while(tab[i] != 0){  
    tab[i]++;  
    i++;  
}
```

Mémoire:

Valeurs	Octets
111 (o)	Octet 1
117 (u)	Octet 2
105 (i)	Octet 3
0	Octet 4
0	Octet 5
	Octet 6
	Octet 7

# Exemple

Programme:

```
char tab [4] = "oui";  
int i = 0;  
while(tab[i] != 0){  
    tab[i]++;  
    i++;  
}
```

Mémoire:

Valeurs	Octets
111 (o)	Octet 1
117 (u)	Octet 2
105 (i)	Octet 3
0	Octet 4
0	Octet 5
	Octet 6
	Octet 7

# Exemple

Programme:

```
char tab [4] = "oui";  
int i = 0;  
while(tab[i] != 0){  
    tab[i]++;  
    i++;  
}
```

Mémoire:

Valeurs	Octets
112 (p)	Octet 1
117 (u)	Octet 2
105 (i)	Octet 3
0	Octet 4
0	Octet 5
	Octet 6
	Octet 7

# Exemple

Programme:

```
char tab [4] = "oui";  
int i = 0;  
while(tab[i] != 0){  
    tab[i]++;  
    i++;  
}
```

Mémoire:

Valeurs	Octets
112 (p)	Octet 1
117 (u)	Octet 2
105 (i)	Octet 3
0	Octet 4
1	Octet 5
	Octet 6
	Octet 7

# Exemple

Programme:

```
char tab [4] = "oui";  
int i = 0;  
while(tab[i] != 0){  
    tab[i]++;  
    i++;  
}
```

Mémoire:

Valeurs	Octets
112 (p)	Octet 1
117 (u)	Octet 2
105 (i)	Octet 3
0	Octet 4
1	Octet 5
	Octet 6
	Octet 7

# Exemple

Programme:

```
char tab [4] = "oui";  
int i = 0;  
while(tab[i] != 0){  
    tab[i]++;  
    i++;  
}
```

Mémoire:

Valeurs	Octets
112 (p)	Octet 1
118 (v)	Octet 2
105 (i)	Octet 3
0	Octet 4
1	Octet 5
	Octet 6
	Octet 7



# Exemple

Programme:

```
char tab [4] = "oui";  
int i = 0;  
while(tab[i] != 0){  
    tab[i]++;  
    i++;  
}
```

Mémoire:

Valeurs	Octets
112 (p)	Octet 1
118 (v)	Octet 2
105 (i)	Octet 3
0	Octet 4
2	Octet 5
	Octet 6
	Octet 7

# Exemple

Programme:

```
char tab [4] = "oui";  
int i = 0;  
while(tab[i] != 0){  
    tab[i]++;  
    i++;  
}
```

Mémoire:

Valeurs	Octets
112 (p)	Octet 1
118 (v)	Octet 2
105 (i)	Octet 3
0	Octet 4
2	Octet 5
	Octet 6
	Octet 7

# Exemple

Programme:

```
char tab [4] = "oui";  
int i = 0;  
while(tab[i] != 0){  
    tab[i]++;  
    i++;  
}
```

Mémoire:

Valeurs	Octets
112 (p)	Octet 1
118 (v)	Octet 2
116 (j)	Octet 3
0	Octet 4
2	Octet 5
	Octet 6
	Octet 7

# Exemple

Programme:

```
char tab [4] = "oui";  
int i = 0;  
while(tab[i] != 0){  
    tab[i]++;  
    i++;  
}
```

Mémoire:

Valeurs	Octets
112 (p)	Octet 1
118 (v)	Octet 2
116 (j)	Octet 3
0	Octet 4
3	Octet 5
	Octet 6
	Octet 7

# Exemple

Programme:

```
char tab [4] = "oui";  
int i = 0;  
while(tab[i] != 0){  
    tab[i]++;  
    i++;  
}
```

Mémoire:

Valeurs	Octets
112 (p)	Octet 1
118 (v)	Octet 2
116 (j)	Octet 3
0	Octet 4
3	Octet 5
	Octet 6
	Octet 7

# La boucle for

- L’instruction **for** (en anglais : “pour”) s’écrit ainsi :

```
for(depart; condition; instruction){  
    code  
}
```

- L’instruction `depart` sera exécutée une fois au début
- Tant que `condition` sera vraie, `code` et `instruction` seront exécutées en boucle.

# Exemple

Programme:

```
char tab [4] = "oui";  
for(int i = 0; tab[i] != 0; i++){  
    tab[i]++;  
}
```

Mémoire:

Valeurs	Octets
	Octet 1
	Octet 2
	Octet 3
	Octet 4
	Octet 5
	Octet 6
	Octet 7

# Exemple

Programme:

```
char tab [4] = "oui";  
for(int i = 0; tab[i] != 0; i++){  
    tab[i]++;  
}
```

Mémoire:

Valeurs	Octets
111 (o)	Octet 1
117 (u)	Octet 2
105 (i)	Octet 3
0	Octet 4
	Octet 5
	Octet 6
	Octet 7



# Exemple

Programme:

```
char tab [4] = "oui";  
for( int i = 0; tab[i] != 0; i++){  
    tab[i]++;  
}
```

Mémoire:

Valeurs	Octets
111 (o)	Octet 1
117 (u)	Octet 2
105 (i)	Octet 3
0	Octet 4
0	Octet 5
	Octet 6
	Octet 7

# Exemple

Programme:

```
char tab [4] = "oui";  
for(int i = 0; tab[i] != 0; i++){  
    tab[i]++;  
}
```

Mémoire:

Valeurs	Octets
111 (o)	Octet 1
117 (u)	Octet 2
105 (i)	Octet 3
0	Octet 4
0	Octet 5
	Octet 6
	Octet 7

# Exemple

Programme:

```
char tab [4] = "oui";  
for(int i = 0; tab[i] != 0; i++){  
    tab[i]++;  
}
```

Mémoire:

Valeurs	Octets
112 (p)	Octet 1
117 (u)	Octet 2
105 (i)	Octet 3
0	Octet 4
0	Octet 5
	Octet 6
	Octet 7

# Exemple

Programme:

```
char tab [4] = "oui";  
for(int i = 0; tab[i] != 0; i++){  
    tab[i]++;  
}
```

Mémoire:

Valeurs	Octets
112 (p)	Octet 1
117 (u)	Octet 2
105 (i)	Octet 3
0	Octet 4
1	Octet 5
	Octet 6
	Octet 7

# Exemple

Programme:

```
char tab [4] = "oui";  
for(int i = 0; tab[i] != 0; i++){  
    tab[i]++;  
}
```

Mémoire:

Valeurs	Octets
112 (p)	Octet 1
117 (u)	Octet 2
105 (i)	Octet 3
0	Octet 4
1	Octet 5
	Octet 6
	Octet 7

# Exemple

Programme:

```
char tab [4] = "oui";  
for(int i = 0; tab[i] != 0; i++){  
    tab[i]++;  
}
```

Mémoire:

Valeurs	Octets
112 (p)	Octet 1
118 (v)	Octet 2
105 (i)	Octet 3
0	Octet 4
1	Octet 5
	Octet 6
	Octet 7

# Exemple

Programme:

```
char tab [4] = "oui";  
for(int i = 0; tab[i] != 0; i++){  
    tab[i]++;  
}
```

Mémoire:

Valeurs	Octets
112 (p)	Octet 1
118 (v)	Octet 2
105 (i)	Octet 3
0	Octet 4
2	Octet 5
	Octet 6
	Octet 7

# Exemple

Programme:

```
char tab [4] = "oui";  
for(int i = 0; tab[i] != 0; i++){  
    tab[i]++;  
}
```

Mémoire:

Valeurs	Octets
112 (p)	Octet 1
118 (v)	Octet 2
105 (i)	Octet 3
0	Octet 4
2	Octet 5
	Octet 6
	Octet 7



# Exemple

Programme:

```
char tab [4] = "oui";  
for(int i = 0; tab[i] != 0; i++){  
    tab[i]++;  
}
```

Mémoire:

Valeurs	Octets
112 (p)	Octet 1
118 (v)	Octet 2
116 (j)	Octet 3
0	Octet 4
2	Octet 5
	Octet 6
	Octet 7

# Exemple

Programme:

```
char tab [4] = "oui";  
for(int i = 0; tab[i] != 0; i++){  
    tab[i]++;  
}
```

Mémoire:

Valeurs	Octets
112 (p)	Octet 1
118 (v)	Octet 2
116 (j)	Octet 3
0	Octet 4
3	Octet 5
	Octet 6
	Octet 7

# Exemple

Programme:

```
char tab [4] = "oui";  
for(int i = 0; tab[i] != 0; i++){  
    tab[i]++;  
}
```

Mémoire:

Valeurs	Octets
112 (p)	Octet 1
118 (v)	Octet 2
116 (j)	Octet 3
0	Octet 4
3	Octet 5
	Octet 6
	Octet 7

# Exemple

Programme:

```
char tab [4] = "oui";  
for(int i = 0; tab[i] != 0; i++){  
    tab[i]++;  
}
```

Mémoire:

Valeurs	Octets
112 (p)	Octet 1
118 (v)	Octet 2
116 (j)	Octet 3
0	Octet 4
3	Octet 5
	Octet 6
	Octet 7

# Durée de validité d'une variable

- Une variable déclarée entre deux accolades existe uniquement entre ces deux accolades
- Une variable déclarée entre les parenthèses d'une boucle for, n'existe que pour cette boucle for

```
int a; // a existe pour toute la durée du programme
if (a > 2){
    int b = 1; // b existe uniquement entre les accolades du if
} else {
    int c = 3; // c existe uniquement entre les accolades du else
}
```

# **Faible de sécurité de type buffer overflow**

- Il est possible de lire ou d'écrire en dehors des limites d'un tableau, cela peut causer des failles de sécurité appelées buffer overflow
- Pour éviter ce problème, il faut toujours contrôler les limites des tableaux avant d'écrire dedans

# Placement des variables dans la mémoire

- Lors des exemples du cours, pour des raisons de simplicité, les variables étaient arrangées dans la mémoire dans l'ordre dans lequel elles étaient déclarées
- Cela n'est pas toujours le cas. Le langage C n'impose pas d'ordre pour les variables dans la mémoire
- Ainsi, pour des raisons de sécurité (évoquées dans la slide précédente) ou de performance, le compilateur est libre de réorganiser la position des variables dans la mémoire.

# Mise en pratique

```
printf("< TP 2 >  
-----  
      \  ^  ^  
        ^  
      (oo)\_____  
      (__) \      ) \/  
           ||-----w ||  
           ||         ||  
")
```