

## Partie 5 : Types structurés-Tableaux

- ▶ Tableaux
- ▶ Chaînes de caractères : principes
- ▶ Arguments de la fonction main

## Rappel : Types de base

- ▶ Entiers (int)
- ▶ Réels (float)
- ▶ Booléens (true/false)
- ▶ Caractère
- ▶ Pointeur (adresse)
- ▶ ...

## Tableaux

Définition :

Un tableau à une dimension

```
<type> <identificateur> [N]
```

est :

- ▶ une structure de données qui contient **N variables de même type**
- ▶ chacune accessible par une clé (ou indice) comprise entre 0 et N-1
- ▶ le nombre d'éléments du tableau (dimension ou taille) est constant.
  - C90 : N est une constantes. C99 : N peut être une expression.

## Tableaux

Définition :

- ▶ l'accès aux éléments s'effectue directement par la clé
- ▶ pour accéder à l'élément d'indice i on écrit T[i].
- ▶ Le **temps de calcul** pour **accéder à un élément du tableau** est **constant**.

## Tableaux : exemples

```
char T[10]; /*tableau T de 10 caracteres.*/  
T[0] = 'a'; /*affectation de la lere case de T.*/
```

	0	1			i				9	
T	a	g	h	d	b	t	v	y	q	u

Que vaut T[3] ?

AlgoProg 2015-16 91

## Tableaux : exemples

Initialiser le tableau T pour que T[i] contienne x :

```
int T[10];  
for(int i = 0; i < 10; i++)  
    T[i] = x;
```

Initialiser le tableau T pour que T[i] contienne i\*i :

```
int T[10];  
for(int i = 0; i < 10; i++)  
    T[i] = i*i;
```

AlgoProg 2015-16 92

## Tableaux

Définition :

Un tableau à k dimensions

<type> <identificateur> [N<sub>1</sub>] [N<sub>2</sub>]...[N<sub>k</sub>]

est :

- ▶ une structure de données qui contient N<sub>1</sub> \* N<sub>2</sub> \*... \* N<sub>k</sub> **variables de même type**
- ▶ Les indices de la j<sup>ème</sup> dimension varient entre 0 et N<sub>j</sub>-1
- ▶ le nombre d'éléments du tableau (dimension ou taille) est constant.
  - C99 : N<sub>j</sub> est une constantes. C99 : N<sub>j</sub> peut être une expression.

AlgoProg 2015-16 93

## Tableaux

Définition :

- ▶ l'accès aux éléments s'effectue directement par les clés

Par exemple :

si T est un tableau à 2 dimensions pour accéder à l'élément d'indice (i, j) on écrit T[i][j].

- ▶ Le **temps de calcul** pour accéder à un élément du tableau est **constant**.

AlgoProg 2015-16 94

## Tableaux : exemples

```
int mat[5][3];  
/* mat est une matrice d'entiers avec 5 lignes et 3  
colonnes.*/  
  
mat[4][2] = 7;  
/*affectation de la "derniere" case de mat.*/
```

mat

	0	1	2
0			
1			
2			
3			
4			7

AlgoProg 2015-16 95

## Tableaux : exemples

Initialiser la matrice 5 x 3 pour que `mat[i][j]` contienne 0

```
int mat[5][3] ;  
for(int i = 0; i < 5;i++)  
    for(int j = 0; j < 3; j++)  
        mat[i][j] = 0;
```

Dans quel ordre se  
remplit le tableau ?

	0	1	2
0			
1			
2			
3			
4			

AlgoProg 2015-16 96

## Tableaux : exemples

Initialiser la matrice 5 x 3 pour que `mat[i][j]` contienne `i+j`

```
int mat[5][3] ;  
for(int i = 0; i < 5;i++)  
    for(int j = 0; j < 3; j++)  
        mat[i][j] = i+j;
```

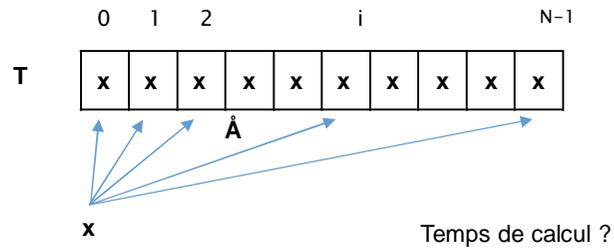
AlgoProg 2015-16 97

## Tableaux : activités

- ▶ Initialiser un tableau.
- ▶ Afficher les éléments d'un tableau.
- ▶ Rechercher si une valeur est dans un tableau.
- ▶ Échanger le contenu de 2 cases du tableau.
- ▶ Insérer un nouvel élément dans un tableau.
- ▶ Pour tous les éléments du tableau faire...
- ▶ Existe-t-il un élément du tableau qui...?
- ▶ **Trier un tableau.**

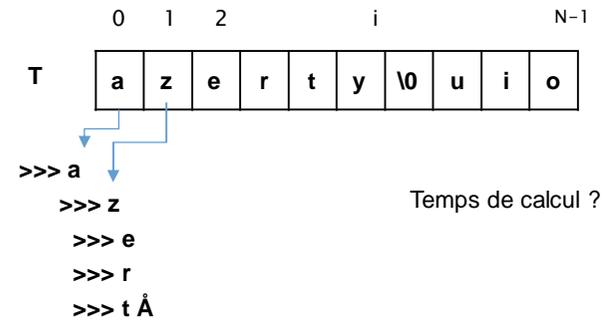
AlgoProg 2015-16 98

## Initialiser un tableau



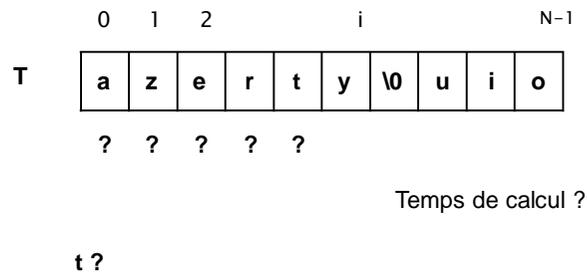
AlgoProg 2015-16 99

## Afficher les éléments d'un tableau



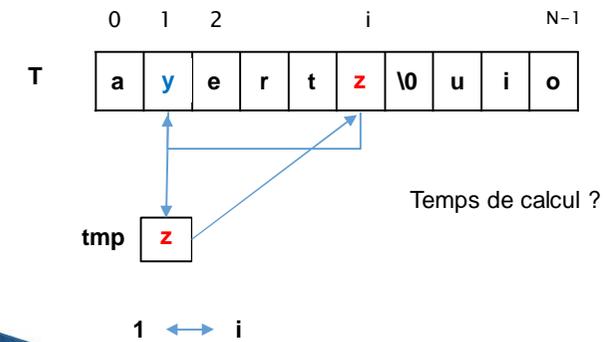
AlgoProg 2015-16 100

## Rechercher un élément dans un tableau



AlgoProg 2015-16 101

## Echange le contenu de 2 cases dans un tableau



AlgoProg 2015-16 102

### Insérer un élément dans un tableau

0 1 2                    i                    N-1

T

b	o	n	o	u	r		u	i	o
---	---	---	---	---	---	--	---	---	---

j

Méthode ?  
Temps de calcul ?

AlgoProg 2015-16 103

### Insérer un élément dans un tableau

0 1 2                    i                    N-1

T

b	o	n	o	u	r		u	i	o
---	---	---	---	---	---	--	---	---	---

j

Méthode ?  
Temps de calcul ?

AlgoProg 2015-16 104

### Insérer un élément dans un tableau

0 1 2                    i                    N-1

T

b	o	n	o	u	r			u	i	o
---	---	---	---	---	---	--	--	---	---	---

j

Méthode ?  
Temps de calcul ?

AlgoProg 2015-16 105

### Insérer un élément dans un tableau

0 1 2                    i                    N-1

T

b	o	n	o	u	r	r		u	i	o
---	---	---	---	---	---	---	--	---	---	---

j

Méthode ?  
Temps de calcul ?

AlgoProg 2015-16 106

### Insérer un élément dans un tableau

0 1 2                    i                    N-1

T

b	o	n	o	u	u	r	\0	i	o
---	---	---	---	---	---	---	----	---	---

Méthode ?

Temps de calcul ?

AlgoProg 2015-16 107

### Insérer un élément dans un tableau

0 1 2                    i                    N-1

T

b	o	n	o	o	u	r	\0	i	o
---	---	---	---	---	---	---	----	---	---

Méthode ?

Temps de calcul ?

AlgoProg 2015-16 108

### Insérer un élément dans un tableau

0 1 2                    i                    N-1

T

b	o	n	j	o	u	r	\0	i	o
---	---	---	---	---	---	---	----	---	---

Méthode ?

Temps de calcul ?

AlgoProg 2015-16 109

### Insérer un élément dans un tableau

0 1 2                    i                    N-1

T

b	o	n	j	o	u	r	\0	i	o
---	---	---	---	---	---	---	----	---	---

Méthode ?

Temps de calcul ?

AlgoProg 2015-16 110

## Chaines de caractères

Les chaînes sont mémorisées dans des tableaux de caractères terminés par `'\0'`

- ▶ Par exemple la chaîne "toto" est mémorisée comme :



- ▶ Un type possible pour représenter les chaînes est `char []` (tableau de caractères).
- ▶ On rencontre aussi `char *` (hors programme).
- ▶ Plusieurs fonctions sont prédéfinies pour les chaînes.

Ex. comparaison : `strcmp`, longueur `strlen`.

```
#include<string.h>
```

AlgoProg 2015-16 111

## Chaines de caractères

La bibliothèque prédéfinit plusieurs fonctions de manipulation des chaînes. Parmi elles :

- ▶ `strlen` : longueur d'une chaîne.
- ▶ `strcmp` (`strncmp`) : comparaison de chaîne
- ▶ `strcat` : concaténation de chaînes
- ▶ `strcpy` (`strncpy`) : copie de chaînes



Ne pas confondre caractères et chaînes de caractères.

```
char s = 'A'  
char *s = 'A'
```

AlgoProg 2015-16 112

## Les arguments du main

```
#include <stdio.h>  
#include<stdlib.h>  
int  
main(void){  
    printf("Bonjour\n");  
    EXIT_SUCCESS;  
}  
>>> gcc -Wall -std=c99 -o salutation salutation.c  
>>> ./salutation  
>>> Bonjour  
>>>
```

AlgoProg 2015-16 113

## Les arguments du main

Comment personnaliser le message affiché ?

```
>>> ./salutation Madame  
>>> Bonjour Madame  
>>>
```

On veut passer des paramètres à l'exécutable  
Donc donner des arguments au main

AlgoProg 2015-16 114

## Les arguments du main

Pour permettre le passage d'arguments à la fonction main, on la définit de la façon suivante :

```
int main (int argc, char *argv[] )
{
    .....
}
```

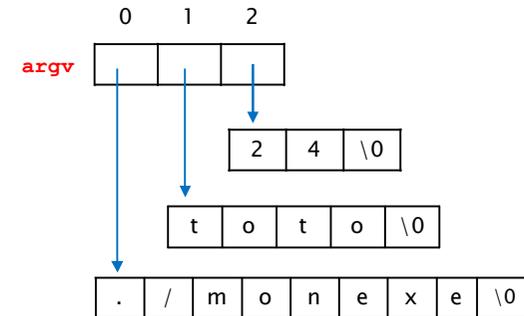
**argc** : initialisé par le nombre d'arguments avec lesquels l'exécutable correspondant est lancé (exécutable compris)

**argv** : tableau de chaînes de caractères initialisé par les arguments avec lesquels l'exécutable correspondant est lancé.

AlgoProg 2015-16 115

## Les arguments du main

Au lancement de la commande shell `./monexe toto 24`



AlgoProg 2015-16 116

## Les arguments du main

```
#include <stdio.h>
#include<stdlib.h>
```

```
int
main(int argc, char *argv[]){
    printf("Bonjour %s\n",argv[1]);
    EXIT_SUCCESS;
}
```

```
>>> gcc -Wall -std=c99 -o salutation salutation.c
```

```
>>> ./salutation Madame
```

```
>>> Bonjour Madame
```

```
>>> ./salutation Paul
```

```
>>> Bonjour Paul
```

AlgoProg 2015-16 117