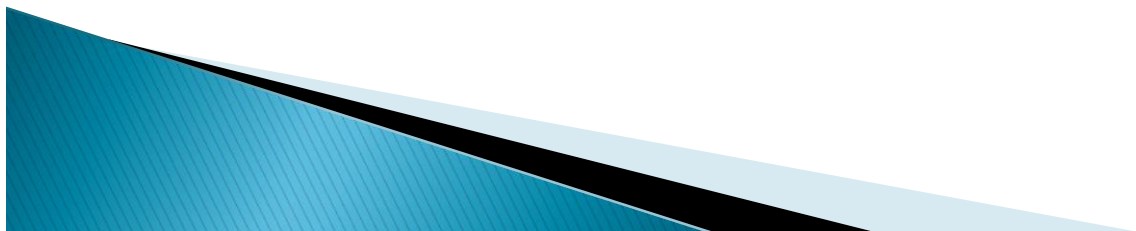


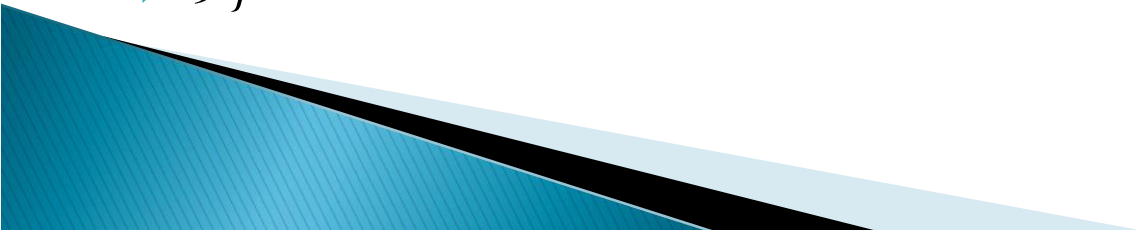
Les Pointeurs

- ▶ Un pointeur est une variable dont la valeur est l'adresse d'une cellule de la mémoire
- ▶ Une adresse est un nombre entier qui donne l'indice d'un élément dans la mémoire de l'ordinateur
- ▶ Un pointeur donne la possibilité de manipuler la variable dont la valeur du pointeur est l'adresse.

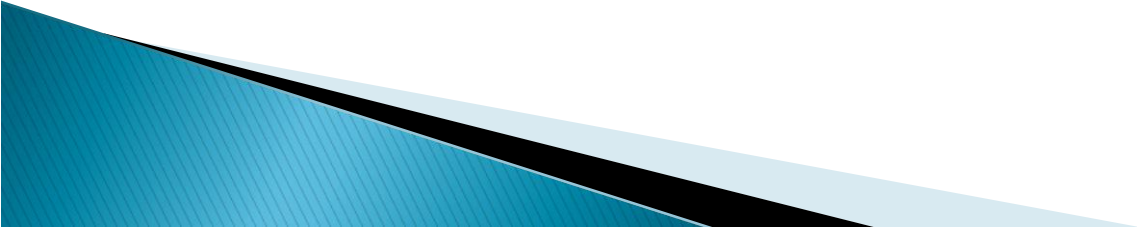


Déclaration et Initialisation de Pointeurs

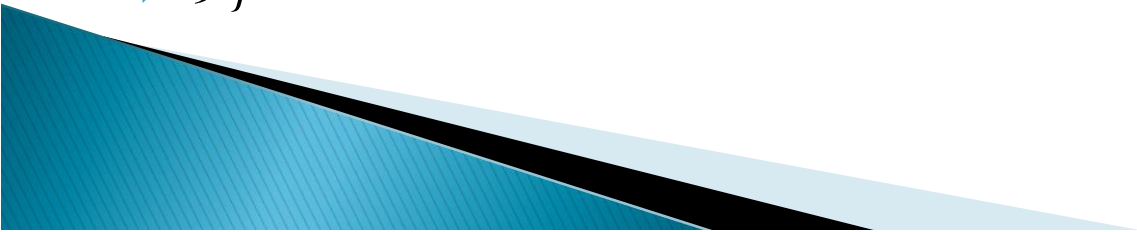
- ▶ Déclaration pointeur typé :
< type > * < identificateur > <= initialisation-opt >;
 - ▶ Déclaration pointeur atypique :
void * < identificateur > <= initialisation-opt >;

 - ▶ *1 #include < stdio.h >*
 - ▶ *2 int main()*
 - ▶ *3 {*
 - ▶ *4 int variable = 1 ;*
 - ▶ *5 int * pt = NULL;*
 - ▶ *6 int ** pt_pt = NULL;*
 - ▶ *7 int **** pt4 ;*
 - ▶ *8 return 0;*
 - ▶ *9 }*
- 

L'opérateur &

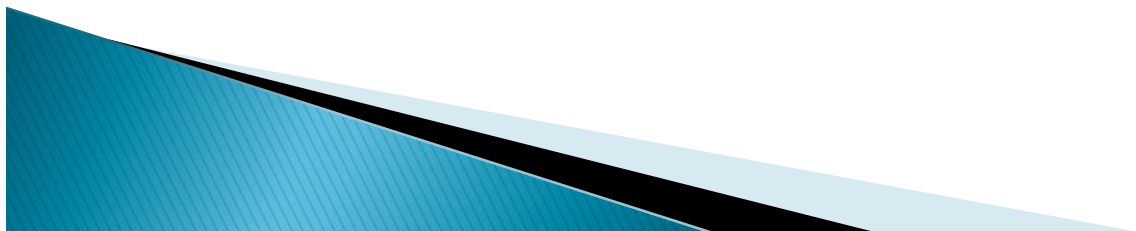
- ▶ L'opérateur & permet de connaître l'adresse d'une variable
 - ▶ Syntaxe : **& < variable >**
 - ▶ *1 #include < stdio.h >*
 - ▶ *2 int main()*
 - ▶ *3 {*
 - ▶ *4 int variable = 1 ;*
 - ▶ *5 int * pt_variable = &variable ;*
 - ▶ *6 int ** pt_pt = &pt_variable ;*
 - ▶ *7 return 0;*
 - ▶ *8 }*
- 

L'opérateur *

- ▶ L'opérateur d'indirection * permet de manipuler l'objet pointé
 - ▶ Syntaxe : * < **expression** >
 - ▶ *1 #include < stdio.h >*
 - ▶ *2 int main()*
 - ▶ *3 {*
 - ▶ *4 int variable = 1 ;*
 - ▶ *5 int *pt_variable = NULL;*
 - ▶ *6 pt_variable = &variable ;*
 - ▶ *7 *pt_variable = 2 ;*
 - ▶ *8 return 0;*
 - ▶ *9 }*
- 

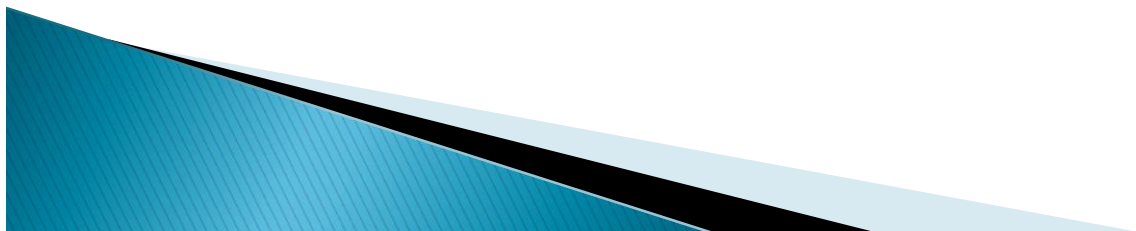
Arithmétique des adresses

- ▶ Soit **<expression>** l'adresse d'un objet O1 de type **Type**
- ▶ **<expression> + 1** : ajoute **sizeof(Type)** à **<expression>**
- ▶ **<expression> + n** correspond à **(Type*)(<expression> + n*sizeof(Type))**
- ▶ Notations équivalentes :
 - ***(<expression> + 1)** et **<expression>[1]**
 - ...
 - ***(<expression> + n)** et **<expression>[n]**



Tableaux et Pointeurs

- ▶ L'identificateur d'un tableau en C est un pointeur vers le premier élément du tableau
- ▶ Considérons la déclaration : **int tab[10];**
- ▶ **tab** est équivalent à **&tab[0]**
- ▶ Equivalence entre :
 - ***tab** et **tab[0]**
 - ***(tab + 1)** et **tab[1]**
 - ***(tab + 2)** et **tab[2]**
 - ...



Passage de paramètres et Pointeurs

- ▶ Exemple : échanger la valeur de deux variables passées en argument

```
1 #include <stdio.h>
2 void echange (int *p, int *q)
3 {
4     int x = *p ;
5     *p = *q ;
6     *q = x ;
7 }

8 int main( )
9 {
10     int a = 2, b = 3;
11     int *p=&a, *q=&b;
12     echange(p,q) ;
13     return 0;
14 }
```

