

Langage C avancé

Les Tableaux

Samuel KOKH

`samuel.kokh@cea.fr`

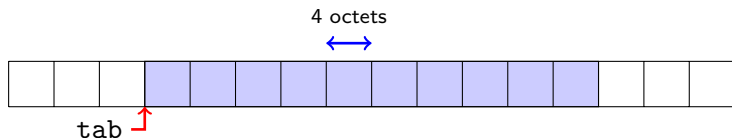
MACS 1 – Institut Galilée

Les tableaux (statiques) en C

Les tableaux n'existent pas (vraiment) en C!

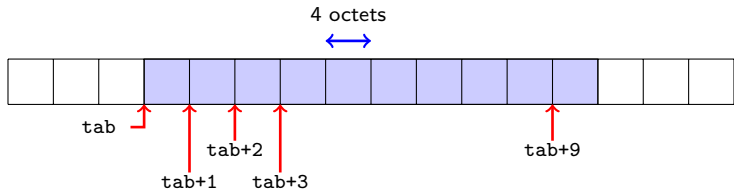
Et pourtant...

```
int tab[10] ; // reserve un espace memoire de taille
              // 10 * sizeof(int) = 40 octets
              // dont l'adresse demarrage est
              // tab
```



Comment accéder à cet espace mémoire ?

Par les opérations classiques sur les pointeurs.

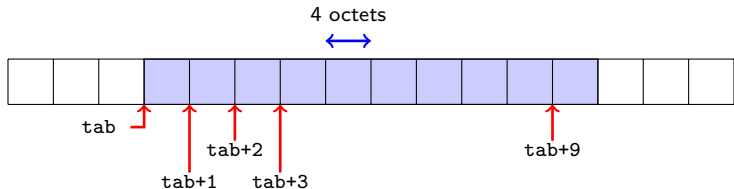


```
int tab[10] ;  
*(tab + 0) = 44;  
*(tab + 1) = 2;  
*(tab + 9) = 17;
```

44	2								17
----	---	--	--	--	--	--	--	--	----

Comment accéder à cet espace mémoire ?

Grâce aux opérateurs []



```
int tab[10] ;  
tab[0] = 44;  
tab[1] = 2;  
tab[9] = 17;
```

44	2									17
----	---	--	--	--	--	--	--	--	--	----

Liens entre les pointeurs et la syntaxe associée aux tableaux

- `tab` : adresse de démarrage de la zone mémoire où sont stockés les éléments du tableau
- `tab ~ &(tab[0])` = adresse du premier élément du tableau
- `(tab+4) ~ &(tab[4])` = adresse du quatrième élément du tableau
- `*(tab+4) ~ tab[4]` = donne accès au quatrième élément du tableau (en lecture/écriture)

Ces deux codes sont équivalents.

```
int *ptr = NULL;
int tab[10];

tab[0] = 1;
tab[4] = 17;
ptr = &(tab[0]);
ptr = &(tab[7]);
printf("%d\n", tab[8]);
```

```
int *ptr = NULL;
int tab[10];

*tab = 1;
*(tab + 4) = 17;
ptr = tab;
ptr = tab + 7;
printf("%d\n", *(tab + 8));
```

Les tableaux sont aussi dangereux que les pointeurs !

Le C ne réalise pas de vérification de dépassement des bornes !

```
int tab[10] ;  
tab[0] = 44;  
tab[1] = 2;  
tab[9] = 17;  
tab[-2] = 7;    // !! danger !!!  
tab[10] = 254 ; // !! danger !!!
```

