



ECE PARIS • LYON
ÉCOLE D'INGÉNIEURS

Algorithmique et programmation structurée en C

TP3

Langage C

Mémoire, adresses, pointeurs

Exercices

Pour les exercices suivants, aidez-vous de schémas.

Exercice 1

Soit le code suivant :

```
int* pt ;  
int val1, val2, val3 ;  
val1 = 10 ;  
val2 = 20 ;  
val3 = 1 ;  
pt = &val1 ;  
*pt = 100 ;
```

Que représente `pt`, `&pt` et `*pt` ? Faire un schéma mémoire de cette séquence de code.

Que représente les instructions suivantes : attention à la priorité de opérations

- ⊙ `*pt + 3`
- ⊙ `pt + 2`
- ⊙ `*pt += 5 ;`
- ⊙ `(*pt)++ ;`
- ⊙ `++*pt ;`
- ⊙ `*(pt + 2)++ ;`
- ⊙ `*pt += val3 ;`
- ⊙ `*(pt + val3) ;`

Exercice 2

- Déclarer un entier val1, un caractère non-signé val2, et un caractère nommé lettre.
- Déclarer un pointeur sur chacune de ces trois variables nommés pt1, pt2 et pt3.
- Déclarer un pointeur sur chacun des 3 pointeurs précédents que vous nommerez ppt1, ppt2 et ppt3.
- Réaliser les affectations permettant de « raccrocher » tous les pointeurs sur leur variable respective.

- ◉ Initialiser les 3 premières variables par une affectation directe et affichez leur contenu à l'écran.
- ◉ Modifier les 3 premières variables en utilisant cette fois une affectation indirecte à 1 indirection.
- ◉ Afficher de nouveau la valeur des 3 variables val1, val2 et lettre en utilisant cette fois la première indirection
- ◉ Modifier les 3 variables par une affectation utilisant la 2ème indirection.
- ◉ Afficher de nouveau la valeur des 3 variables val1, val2 et lettre en utilisant cette fois la deuxième indirection.

- ◉ Faire un schéma mémoire de toutes les étapes de cet exercice.