

Un pointeur est une variable dont la valeur est une adresse mémoire. Plus précisément, la déclaration d'une variable pointeur fait apparaître le type de l'objet pointé (ex: adresse d'un **int**, adresse d'un **float**, etc...).

Un **tableau à-la-C** occupe un bloc compact d'espace mémoire suffisant pour contenir la valeur de tous ses éléments.

Contrairement aux variables (classique ou pointeur) **le nom du tableau, utilisé seul dans une expression, est une constante**: c'est l'adresse du premier élément du tableau.

Dans l'illustration suivante, nous avons des variables de type **int**, des variables pointeurs de type **int*** et un tableau de 2 **int**.

Pour simplifier vous indiquerez les valeurs entières en décimal et les adresses en hexadécimal. Complétez le contenu de la mémoire après les déclarations et instructions ci-dessous et ensuite donnez la valeur des expressions apparaissant dans les questions de 1) à 10) *en supposant qu'elles exécutées dans cet ordre*. Mettez à jour l'état de la mémoire si nécessaire.

Illustrez aussi l'exercice en utilisant les conventions graphiques « boîte + flèche »

```
int v1 = 19;
int v2 = 50 ;
int* v3 = &v1 ;
int* v4 = &v2 ;
int v5[2] = {-13, 27};
```

déclarations	Adresse en hexa	mémoire 32 bits
	
int v1	04	
int v2	08	
int* v3	0C	
int* v4	10	
int v5[2]	14	
	18	
	

Questions	Réponses : valeur en decimal, adresse en hexa
1) valeur de l'expression: v1	
2) valeur de l'expression: v3	
3) valeur de: *v3 v3[0]	
4) valeur de l'expression: &v3	
5) valeur de l'expression: v5	
6) valeur de l'expression: v5[0]	
7) valeur de l'expression: &v5[1]	
8) valeur de l'expression: *v4 + 2	
9) valeur de l'expression: (*v4)++	
10) valeur de l'expression: *v3= *v4 + 3	