

**Rappel:** En langage C++ une variable possède un **type** qui détermine le nombre d'octets qu'elle occupe en mémoire pour stocker sa **valeur**.

L' **adresse** d'une variable est l'adresse de son premier octet

Grâce à l'**identificateur** de la variable (son nom) on peut lire sa valeur ou la modifier sans connaître explicitement son adresse. En effet ce sont des traitements ultérieurs à l'écriture du programme qui décident de sa position en mémoire au moment de l'exécution. Malgré cela il est très important de maîtriser le concept d'adresse pour la suite du cours.

A **droite** d'un opérateur '=' l'**identificateur** est équivalent à "**la valeur de la variable**".

Le C++ définit par ailleurs les 2 opérateurs '&' et '\*', utilisés comme suit :

**&identificateur** est équivalent à l' "**adresse de la variable identificateur**".

**\*adresse** est équivalent à la "**valeur de la mémoire à l'adresse adresse**".

Dans l'illustration suivante, on travaille avec une machine 32 bits. Nous avons quatre variables de type entier sur 4 octets, d'identificateurs : v1, v2, v3, v4 . Pour simplifier, la valeur indiquée en mémoire est en décimal bien qu'elle ne soit en réalité qu'une suite de 0 et de 1.

identificateurs des variables	Adresse 32 bits en hexa	mémoire 32 bits
	.....	<i>Valeur indiquée en décimal</i>
v1	00 00 00 04	12 <sub>10</sub>
v2	00 00 00 08	77 <sub>10</sub>
v3	00 00 00 0C	-5 <sub>10</sub>
v4	00 00 00 10	99 <sub>10</sub>
	00 00 00 14	
	00 00 00 18	
	.....	

**Questions**

**Réponses :** *valeur en decimal, adresse en hexa*

1) valeur de v1, v2, v3, v4

2) adresse de v1 (en hexa)

3) &v3

4) valeur en mémoire à l'adresse &v3

5) \*(&v4)

6) &v1 + 4 octets

7) \*(&v1 + 8 octets)