

Exercice 1

```
main()
{
    int    A = 1;
    int    B = 2;
    int    C = 3;
    int    *P1, *P2;
    P1=&A;
    P2=&C;
    *P1=(*P2)++;
    P1=P2;
    P2=&B;
    *P1-=*P2;
    ++*P2;
    *P1*-=*P2;
    A=++*P2**P1;
    P1=&A;
    *P2=*P1/=*P2;
    return 0;
}
```

Copiez le tableau suivant et complétez-le pour chaque instruction du programme ci-dessus.

	A	B	C	P1	P2
Init.	1	2	3	/	/
P1=&A	1	2	3	&A	/
P2=&C					
*P1=(*P2)++					
P1=P2					
P2=&B					
*P1-=*P2					
++*P2					
P1-=*P2					
A=++*P2**P1					
P1=&A					
*P2=*P1/=*P2					

Exercice 2

Soit P un pointeur qui 'pointe' sur un tableau A :

```
int A[] = {12, 23, 34, 45, 56, 67, 78, 89, 90};
```

```
int *P;
```

```
P = A;
```

Quelles valeurs ou adresses fournissent ces expressions :

- *P+2
- *(P+2)
- &P+1
- &A[4]-3
- A+3
- &A[7]-P

- g) $P+(*P-10)$
- h) $*(P+*(P+8)-A[7])$

Exercice 3

Ecrire un programme qui range les éléments d'un tableau A du type int dans l'ordre inverse. Le programme utilisera des pointeurs P1 et P2 et une variable numérique AIDE pour la permutation des éléments.

Exercice 4

Ecrire de deux façons différentes, un programme qui vérifie sans utiliser une fonction de <string>, si une chaîne CH introduite au clavier est un palindrome:

- a) en utilisant uniquement le formalisme tableau
- b) en utilisant des pointeurs au lieu des indices numériques

Rappel : Un palindrome est un mot qui reste le même qu'on le lise de gauche à droite ou de droite à gauche :

Exemples : OMAR ==> n'est pas un palindrome

RADAR ==> est un palindrome