

### Exercice 1

	A	B	C	P1	P2
Init.	1	2	3	/	/
P1=&A	1	2	3	&A	/
P2=&C	1	2	3	&A	&C
*P1=(*P2)++	3	2	4	&A	&C
P1=P2	3	2	4	&C	&C
P2=&B	3	2	4	&C	&B
*P1-=*P2	3	2	2	&C	&B
++*P2	3	3	2	&C	&B
*P1*=*P2	3	3	6	&C	&B
A=++*P2**P1	24	4	6	&C	&B
P1=&A	24	4	6	&A	&B
*P2=*P1/=*P2	6	6	6	&A	&B

### Exercice 2

- a) \*P+2                   => la valeur 14
- b) \*(P+2)               => la valeur 34
- c) &P+1                 => l'adresse du pointeur derrière le pointeur P  
(rarement utilisée)
- d) &A[4]-3             => l'adresse de la composante A[1]
- e) A+3                  => l'adresse de la composante A[3]
- f) &A[7]-P             => la valeur (indice) 7
- g) P+(\*P-10)          => l'adresse de la composante A[2]
- h) \*(P+\*(P+8)-A[7]) => la valeur 23

### Exercice 3

```
#include <stdio.h>
main()
{
    /* Déclarations */
    int A[50]; /* tableau donné */
    int N; /* dimension du tableau */
    int AIDE; /* pour la permutation */
    int *P1, *P2; /* pointeurs d'aide */

    /* Saisie des données */
    printf("Dimension du tableau (max.50) : ");
    scanf("%d", &N);
    for (P1=A; P1<A+N; P1++)
    {
        printf("Elément %d : ", P1-A);
        scanf("%d", P1);
    }
}
```

```
/* Affichage du tableau */
for (P1=A; P1<A+N; P1++)
    printf("%d ", *P1);
    printf("\n");

/* Inverser la tableau */
for (P1=A, P2=A+(N-1); P1<P2; P1++, P2--)
{
    AIDE = *P1;
    *P1 = *P2;
    *P2 = AIDE;
}

/* Edition du résultat */
for (P1=A; P1<A+N; P1++)
    printf("%d ", *P1);
    printf("\n");
return 0;
}
```

#### Exercice 4

a) en utilisant uniquement le formalisme tableau

```
#include <stdio.h>
main()
{
    char CH[101]; /* chaîne donnée */
    int I, J; /* indices courants */
    int PALI; /* indicateur logique: vrai si CH est un palindrome */

    /* Saisie des données */
    printf("Entrez une ligne de texte (max.100 caractères) :\n");
    gets(CH);
    /* Placer J sur la dernière lettre de la chaîne */
    for (J=0; CH[J]; J++)
        ;
    J--;

    /* Contrôler si CH est un palindrome */
    PALI=1;
    for (I=0; PALI && I<J; I++, J--)
        if (CH[I] != CH[J])
            PALI=0;

    /* Affichage du résultat */
    if (PALI) printf("La chaîne \"%s\" est un palindrome.\n", CH);
    else printf("La chaîne \"%s\" n'est pas un palindrome.\n", CH);

    return 0;
}
```

b) en utilisant des pointeurs au lieu des indices numériques :

```
#include <stdio.h>
main()
{
    /* Déclarations */
    char CH[101]; /* chaîne donnée */
    char *P1,*P2; /* pointeurs d'aide */
    int PALI; /* indicateur logique: vrai si CH est un palindrome */

    /* Saisie des données */
    printf("Entrez une ligne de texte (max.100 caractères):\n");
    gets(CH);

    /* Placer P2 sur la dernière lettre de la chaîne */
    for (P2=CH; *P2; P2++)
        ;
    P2--;

    /* Contrôler si CH est un palindrome */
    PALI=1;
    for (P1=CH ; PALI && P1<P2 ; P1++,P2--)
        if (*P1 != *P2) PALI=0;

    /* Affichage du résultat */
    if (PALI)      printf("La chaîne \"%s\" est un palindrome.\n", CH);
    else          printf("La chaîne \"%s\" n'est pas un palindrome.\n", CH);

    return 0;
}
```