R1.08 Informatique 2022 – 2023	TD n°6 Les pointeurs (correction)	UVSQ université PARIS-SACLAY — IUT de Vélizy-Rambouillet CAMPUS DE VÉLIZY-VILLACOUBLAY

Exercice 1 :

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 :

a) ***P+2**
b) ***(P+2)**
c) **&P+1**
d) **&A[4]-3**
e) **A+3**
f) **&A[7]-P**
g) **P+(*P-10)**
h) ***(P+*(P+8)-A[7])**

Voici les valeurs ou adresses fournies par les expressions données dans l'exercice :

a) *P+2 : 34 (la valeur de l'élément à l'adresse P plus 2)
b) *(P+2) : 34 (la valeur de l'élément à l'adresse P+2)
c) &P+1 : l'adresse de P+1 (l'adresse suivant celle de P)
d) &A[4]-3 : l'adresse de A[1] (l'adresse de A[4] moins 3)
e) A+3 : l'adresse de A[3] (l'adresse de A plus 3)
f) &A[7]-P : 3 (l'adresse de A[7] moins l'adresse de A[0])
g) P+(*P-10) : l'adresse de A[2] (l'adresse de A[0] plus la valeur de A[0] moins 10)
h) (P+(P+8)-A[7]) : 12 (la valeur de l'élément à l'adresse P+*(P+8)-A[7])

Note : pour évaluer ces expressions, il est important de se rappeler que P est un pointeur vers un entier et qu'A est un tableau d'entiers. Les opérations arithmétiques sur les pointeurs sont interprétées comme des déplacements d'adresses, tandis que les opérations arithmétiques sur les entiers sont interprétées comme des opérations sur les valeurs des entiers.

Exercice 2 :

Écrire une fonction swap qui a comme paramètres deux pointeurs vers des entiers et qui échange le contenu des deux entiers pointés. Tester cette fonction en écrivant un programme qui échange le contenu de deux entiers a et b en appelant cette fonction.

```
#include <stdio.h>

void swap(int *a, int *b) {
   int temp = *a;
   *a = *b;
   *b = temp;
}

int main(void) {
   int a = 5, b = 10;
   printf("avant: a = %d, b = %d\n", a, b);
   swap(&a, &b);
   printf("après: a = %d, b = %d\n", a, b);
   return 0;
}
```

Exercice 3 :

Écrire une fonction qui a comme paramètres un tableau d'entiers de taille quelconque, la taille du tableau, et 2 pointeurs vers des entiers min et max. La fonction doit renvoyer dans les entiers pointés par min et max respectivement les plus petits et les plus grands entiers du tableau.

```
#include <stdio.h>

void min_max(int *tab, int size, int *min, int *max) {
   *min = *max = tab[0];
   for (int i = 1; i < size; i++) {
      if (tab[i] < *min) *min = tab[i];
      if (tab[i] > *max) *max = tab[i];
   }
}

int main(void) {
   int tab[] = {3, 5, 1, 7, 2, 4, 9, 6, 8, 0};
   int size = sizeof(tab) / sizeof(tab[0]);
   int min, max;
   min_max(tab, size, &min, &max);
   printf("min = %d, max = %d\n", min, max);
   return 0;
}
```

Exercice 4 :

Écrire une fonction qui a en paramètre une chaîne de caractères et qui renvoie par un return le nombre d'occurrences de la lettre 'A'. Cette fonction devra parcourir la chaîne en utilisant un pointeur. Tester cette fonction.

```
#include <stdio.h>

int count_a(char *s) {
   int count = 0;
   for (; *s; s++) {
      if (*s == 'A') count++;
   }
   return count;
}

int main(void) {
   char s[] = "L'ARC-EN-CIEL";
   printf("%d\n", count_a(s));
   return 0;
}
```

Exercice 5 :

Ecrire un programme qui lit un entier X et un tableau A du type **int** au clavier et élimine toutes les occurrences de X dans A en tassant les éléments restants. Le programme utilisera les pointeurs P1 et P2 pour parcourir le tableau.

```
#include <stdio.h>

void remove_x(int x, int *a, int size) {
   int *p1 = a, *p2 = a;
   for (; p2 < a + size; p2++) {
      if (*p2 != x) {
         *p1 = *p2;
         p1++;
      }
   }
   while (p1 < a + size) *p1++ = 0;
}

int main(void) {
   int x, a[10], size;
   printf("Entrez un entier X : ");
   scanf("%d", &x);
   printf("Entrez la taille du tableau A : ");
   scanf("%d", &size);
   printf("Entrez les éléments du tableau A :\n");
   for (int i = 0; i < size; i++) scanf("%d", &a[i]);
   remove_x(x, a, size);
   printf("Tableau A après suppression de X :\n");
   for (int i = 0; i < size; i++) printf("%d ", a[i]);
   printf("\n");
   return 0;
}
```