

Exercice 1. Qu'affiche le programme suivant :

```
#include <stdlib.h>
#include <stdio.h>
void f(int a, int b) {
    int temp = a;
    a = b;
    b = temp;
}
void g(int *pa, int *pb) {
    int temp = *pa;
    *pa = *pb;
    *pb = temp;
}
int main () {
    int x = 9;
    int y = 54;
    printf ("x = %d, y = %d\n", x, y);
    f(x, y);
    printf ("apres appel a la fonction f :\tx = %d, y = %d\n", x, y);
    g(&x, &y);
    printf ("apres appel a la fonction g :\tx = %d, y = %d\n", x, y);
    return EXIT_SUCCESS;
}
```

Exercice 2.

Ecrire une fonction `permutation_circulaire` permettant de modifier les valeurs de 3 variables de telle sorte qu'après l'appel la première variable a pris la valeur de la deuxième variable, la deuxième a pris la valeur de la troisième, et la troisième a pris la valeur qu'avait la première.

Par exemple, si avant l'appel à la fonction `permutation_circulaire` les variables `a`, `b` et `c` ont respectivement pour valeurs 12, 32 et 45, après l'appel elles ont respectivement pour valeurs 32, 45 et 12.

Exercice 3. On considère la fonction `chaine_copie`. Expliquez son fonctionnement.

```
char *
chaine_copie(char *dest, const char *src)
{
    char *tmp = dest;
    while ((*tmp++ = *src++) != '\0')
        ;
    return dest;
}
```

Exercice 4. Ecrire la fonction `char * car_dans_chaine(const char *s, int c)` qui retourne l'adresse de la première occurrence du caractère `c` dans la chaîne `s` ou `NULL` si le caractère n'a pas été trouvé.

Cette fonction existe dans la bibliothèque C standard (`strchr`).

Exercice 5. Utiliser la fonction `strchr` pour écrire la fonction `int compter_occurrences_car(const char *s, int c)` qui retourne le nombre d'occurrences du caractère `c` dans la chaîne `s`.

Exercice 6. Écrire une fonction `char *chaine_concatenation(char *dest, const char *src)` qui concatène deux chaînes de caractères. Elle ajoute la chaîne `src` à la fin de la chaîne `dest` en écrasant le caractère `'\0'` à la fin de `dest`, puis en ajoutant un nouveau caractère `'\0'` final. Les chaînes ne doivent pas se chevaucher, et la chaîne `dest` doit être assez grande pour accueillir le résultat. Cette fonction renvoie un pointeur sur la chaîne destination `dest`.

Ecrire une version utilisant les pointeurs.

Cette fonction existe dans la bibliothèque C standard (`strcat`). Consulter la documentation de cette fonction.

Exercice Complémentaire 1. Ecrire la fonction `char * dernier_car_dans_chaine(const char *s, int c)` qui retourne l'adresse de la dernière occurrence du caractère `c` dans la chaîne `s` ou `NULL` si le caractère n'a pas été trouvé.

Cette fonction existe dans la bibliothèque C standard (`strrchr`).

Exercice Complémentaire 2. Utiliser la fonction `strchr` pour écrire la fonction `void remplacer_occurrences_car(char *s, int c1, int c2)` qui remplace toutes les d'occurrences du caractère `c1` par le caractère `c2` dans la chaîne `s`.

Exercice Complémentaire 3. Analyser chacune des fonctions suivantes (on précisera la forme de l'appel de la fonction) :

```
int f1(int *i) {
    return *i+1;
}

int f2(int *i) {
    return ++(*i);
}

int f3(int *i) {
    return (*i)++;
}

int f4(int *i) {
    printf("%d\n",(*i)++);
    return *i;
}

int f5(int *i) {
    printf("%d\n",*i==0);
    return i;
}

int f6(int *i) {
    printf("%d\n",*i=0);
    return **i;
}
```

Exercice Complémentaire 4. Analyser chacune des fonctions suivantes :

```
void f1(int i,int j) {
    i=j;
}

void f2(int *i,int *j) {
    i=j;
}

void f3(int *i,int *j) {
    *i=*j;
    *j=*i;
}

void f4(int **i,int **j) {
    **i=**j;
    *j=*i;
}
```