

Langage C – TP C 5 - correction exercice 1 question 1

```
#include <stdio.h>
```

```
#define NB_ELEMENTS 100
```

```
void main(void)
```

```
{
```

```
    char tableau_caractere_1[NB_ELEMENTS]="un message", tableau_caractere_2[NB_ELEMENTS];
```

```
    int i=0;
```

```
    while(tableau_caractere_1[i]!='\0')
```

```
    {
```

```
        // facultative
```

```
        i++;
```

```
    }
```

```
        // facultative
```

```
    printf("%s a %d caractere(s)\n",tableau_caractere_1,i);
```

```
    printf("Veuillez, s'il vous plait, saisir un mot : ");
```

```
    scanf("%s",tableau_caractere_2); // pas de caractère & avant une chaîne de caractères
```

```
    for(i=0;tableau_caractere_2[i]!='\0';i++); // for(i=0;tableau_caractere_2[i]!='\0';i++)
```

```
        // {
```

```
        // }
```

```
    printf("%s a %d caractere(s)\n",tableau_caractere_2,i);
```

```
}
```

Langage C – TP C 5 - correction exercice 1 question 1

```
#include <stdio.h>
```

```
#define NB_ELEMENTS 100
```

```
int nombre_caracteres(char*);
```

```
void main(void)
```

```
{
```

```
    char tableau_caractere_1[NB_ELEMENTS]="un message", tableau_caractere_2[NB_ELEMENTS];
```

```
    printf(" %s a %d caractere(s)\n",tableau_caractere_1,nombre_caracteres(tableau_caractere_1));
```

```
    printf("Veuillez, s'il vous plait, saisir un mot : ");
```

```
    scanf("%s",tableau_caractere_2);
```

```
    printf(" %s a %d caractere(s)\n",tableau_caractere2,nombre_caracteres(tableau_caractere_2));
```

```
}
```

```
/* fonction qui renvoie le nombre de caractères d'une chaîne de caractères passée en paramètre */
```

```
int nombre_caracteres(char* tab_car)
```

```
{
```

```
    int i;
```

```
    for(i=0;tab_car[i]!='\0';i++);
```

```
    // int i=0;
```

```
    // while(tab_car[i]!='\0')
```

```
    // {                                // facultative car une seule instruction
```

```
    //     i++;
```

```
    // }                                // facultative
```

```
    return i;
```

```
}
```

Langage C – TP C 5 - correction **exercice 1 question 2**

/ procédure qui affiche à l'envers une chaîne de caractères
passée en paramètre sans en modifier le contenu */*

```
void afficher_envers(char* tab_car)
{
    int i;
    for(i=nombre_caracteres(tab_car)-1;i>=0;i--)
    {
        // facultative car une seule instruction
        printf("%c",tab_car[i]);
    }
    // facultative
    printf("\n");
}
```

Langage C – TP C 5 - correction **exercice 1 question 3**

```
#include <stdio.h>
#define NB_ELEMENTS 100
/* fonction qui a en paramètre deux chaînes de caractères, qui copie la seconde dans la
   première puis renvoie la première chaîne au programme principal pour affichage */
char* copier(char*,char*);

void main(void)
{
    char tab_car_1[NB_ELEMENTS]="ab", tab_car_2[NB_ELEMENTS]="cdefg";
    // tab_car_1 'a' 'b' '\0' ...          tab_car_2 'c' 'd' 'e' 'f' 'g' '\0' ...
    printf("%s",copier(tab_car_1,tab_car_2)); // affiche cdefg
    // tab_car_1 'c' 'd' 'e' 'f' 'g' '\0' ... tab_car_2 inchangé

    char tab_car_3[NB_ELEMENTS]="abcde", tab_car_4[NB_ELEMENTS]="fg";
    // tab_car_3 'a' 'b' 'c' 'd' 'e' '\0' ... tab_car_4 'f' 'g' '\0' ...
    printf("%s",copier(tab_car_3,tab_car_4)); // affiche fg
    // tab_car_3 'f' 'g' '\0' ...          tab_car_4 inchangée
}
```

Langage C – TP C 5 - correction **exercice 1 question 3**

/* fonction qui a en paramètre deux chaînes de caractères, qui copie la seconde dans la première puis renvoie la première chaîne au programme principal pour affichage */

```
char* copier(char* tab_car_1, char* tab_car_2)
{
    int i;
    /* copie des caractères, un à un, de la deuxième chaîne dans la première chaîne à partir du début */
    for(i=0; i<nombre_caracteres(tab_car_2); i++)
    {
        // facultative car une seule instruction
        tab_car_1[i]=tab_car_2[i];
    }
    // facultative
    // copie du caractère '\0' dans la première chaîne après le dernier caractère copié
    tab_car_1[i]='\0';
    return tab_car_1;
}
```

Langage C – TP C 5 - correction **exercice 1 question 4**

```
#include <stdio.h>
#define NB_ELEMENTS 100
/* fonction qui a en paramètre deux chaînes de caractères, qui copie la seconde à la fin
   de la première puis renvoie la première chaîne au programme principal pour
   affichage */
char* concatener(char*,char*);

void main(void)
{
    char tab_car_1[NB_ELEMENTS]="abcde", tab_car_2[NB_ELEMENTS]="fg";
    // tab_car_1 'a' 'b' 'c' 'd' 'e' '\0' ...   tab_car_2 'f' 'g' '\0' ...
    printf("%s",concatener(tab_car_1,tab_car_2)); // affiche abcdefg
    // tab_car_1 'a' 'b' 'c' 'd' 'e' 'f' 'g' '\0' ...   tab_car_2 inchangé
}
```

Langage C – TP C 5 - correction **exercice 1 question 4**

/* fonction qui a en paramètre deux chaînes de caractères, qui copie la seconde à la fin de la première puis renvoie la première chaîne au programme principal pour affichage */

```
char* concatener(char* tab_car_1, char* tab_car_2)
{
    int i=nombre_caracteres(tab_car_1),j;
    /* copie des caractères, un à un, de la deuxième chaîne dans la première chaîne à partir de '\0' */
    for(j=0;j<nombre_caracteres(tab_car_2);j++)
    {
        // facultative car une seule instruction
        tab_car_1[i+j]=tab_car_2[j];
    }
    // facultative
    // copie du caractère '\0' dans la première chaîne après le dernier caractère copié
    tab_car_1[i+j]='\0';
    return tab_car_1;
}
```

Langage C – TP C 5 - correction **exercice 1 question 5**

```
#include <stdio.h>
```

```
#define NB_ELEMENTS 100
```

```
/* fonction qui inverse l'ordre des caractères d'une chaîne de caractères passée  
   en paramètre puis renvoie le résultat au programme principal pour  
   affichage */
```

```
char* inverser(char*);
```

```
void main(void)
```

```
{
```

```
    char tableau_caractere[NB_ELEMENTS]="abcdef";
```

```
    // tableau_caractere 'a' 'b' 'c' 'd' 'e' 'f' '\0' ...
```

```
    inverser(tableau_caractere);
```

```
    // tableau_caractere 'f' 'e' 'd' 'c' 'b' 'a' '\0' ...
```

```
}
```


Langage C – TP C 5 - correction **exercice 1 question 5**

/* fonction qui inverse l'ordre des caractères d'une chaîne de caractères passée en paramètre puis renvoie le résultat au programme principal pour affichage */

```
char* inverser(char* tab_car)
{
    int i;
    char tmp;
    for(i=0;i<nombre_caracteres(tab_car)/2;i++)
    {
        tmp=tab_car[i];
        tab_car[i]=tab_car[nombre_caracteres(tab_car)-1-i];
        tab_car[nombre_caracteres(tab_car)-1-i]=tmp;
    }
    return tab_car;
}
```