

Initiation à la programmation en C

Correction du TP n°5

Antoine Miné

15 mars 2007

Site du cours: <http://www.di.ens.fr/~mine/enseignement/prog2006/>

Exercice 1. Palindromes.

```
#include <string.h>

/* renvoie 1 si s est un palindrome, 0 sinon */
int palindrome(char* s)
{
    /* strlen étant coûteux, on ne l'appelle qu'une fois et on stocke son résultat
       dans une variable */
    int n = strlen(s);
    int i = 0;
    while ( i < n/2 ) {
        /* ATTENTION: l'indice symétrique de i est n-i-1, pas n-i! */
        if ( s[i] != s[n-i-1] ) return 0; /* echec! */
        i++;
    }
    return 1; /* OK */
}
```

Exercice 2. strcat

```
void my_strcat(char* dest, const char* src)
{
    while (*dest) dest++;
    while (*src) *(dest++) = *(src++); /* copie src */
    *dest = 0; /* ajoute le zéro final */
}
```

Exercice 3. Débogage.

Le pointeur `p` n'est pas initialisé, ce qui peut provoquer un `Segmentation fault` dans l'instruction `tab[mid]`. L'appel à `recherche(x, 13, 6)` provoque une boucle infinie : il faut changer `min = mid` et `max = mid` respectivement en `min = mid+1` et `max = mid-1`.

Exercice 4. Insertion.

```
void insert(char *dest, const char *src, int pos)
{
```

```

int ns = strlen(src);
int nd = strlen(dest);
int i;
assert(pos>=0 && pos<=nd); /* vérification */

/* on décale la fin de dest de ns positions */
/* la copie se fait de droite à gauche */
/* on part de l'indice nd pour copier aussi le 0 final */
for ( i=nd; i>=pos; i--)
    dest[i+ns] = dest[i];

/* on copie src au bon endroit dans dest */
for ( i=0; i<ns; i++ )
    dest[i+pos] = src[i];
}

```

Exercice 5. Échappement.

```

/* ATTENTION:
on suit ici la convention de strncpy: un 0 final est toujours ajouté
(contrairement à strncpy) mais il n'est pas compté dans les max
caractères => au plus max+1 caractères sont donc écrits dans dest.
*/
void echappe(char *dest, const char *src, int max)
{
    while ( *src ) {

        if ( max-- <= 0 ) break; /* quitte si dest est plein */

        if ( *src == 13 ) {
            /* \n */
            *(dest++) = '\\';
            if ( max-- <= 0 ) break;
            *(dest++) = 'n';
        }
        else if ( *src >= 32 && *src <= 126 ) {
            /* caractère affichable */
            *(dest++) = *src;
        }
        else {
            /* séquence \xxx en octal */
            unsigned char c = *src; /* conversion en un nombre entre 0 et 255 */
            *(dest++) = '\\';
            if ( max-- <= 0 ) break;
            *(dest++) = '0' + ( c / 64 );
            if ( max-- <= 0 ) break;
            *(dest++) = '0' + ( ( c / 8 ) % 8 );
            if ( max-- <= 0 ) break;
            *(dest++) = '0' + ( c % 8 );
        }

        src++; /* passe au caractère suivant de src */
    }

    *dest = 0;
}

```
