

FIG. 4.1 – Algorithme (en C) de recherche de motif naïf (motif.c)

```

#include <stdio.h>
#include <string.h>
int motif(char * pattern, char * texte) {
    // Les positions sont comptées comme en Pascal : de 1 à la
    // longueur, et non pas à partir de 0
    int i, j ;
    int lp, lt ;           // longueurs des chaînes
    // On vérifie qu'aucune chaîne n'est vide
    if (pattern[0]=='\0' || texte[0]=='\0') return -1 ;
    // Le texte doit être plus long que le pattern
    lp = strlen(pattern) ; lt = strlen(texte) ;
    if (lp > lt) return -2 ;

    i=0 ;
    do {
        for (j=0 ; j < lp && pattern[j]==texte[i+j] ; j++)
            ;
        i++ ;
    } while (i<=(lt-lp) && j < lp) ;
    if (j < lp) return 0 ;
    else return i ;
}

main(int argc, char * argv[]) {
    if (argc > 2) {
        printf("Recherche de |%s| dans |%s| : position : %d\n",
            argv[1], argv[2], motif(argv[1], argv[2])) ;
    }
    else exit (-1) ;
}
    
```

```

mot   T O T O T I   N = 6
facteur T O T I     M = 4

i=1  T O T O T I
j=1  T       Idem
j=2  O       Idem
j=3  T       Idem
j=4  I       Clash

i=2  T O T O T I
j=1  T       Clash

i=3  T O T O T I
j=1  T       Idem
j=2  O       Idem
j=3  T       Idem
j=4  I       Idem
j=5  j=M+1   Trouvé
    
```

FIG. 4.2 – Exemple recherche de facteur linéaire