

TD2

Exercice 1 :

Ecrire un Algo qui lit une chaîne de caractères CH et détermine la longueur de la chaîne à l'aide d'un pointeur P. L'algo n'utilisera pas de variables numériques.

Algorithme longueur de chaîne de caractère :

```
Char CH[100] ;  
Char *p ;  
  
Entier L ;  
  
Début :  
    Afficher("Entrez maximum 100 caractères") ;  
    Lire(CH) ;  
  
    Pour (p=CH ; *p ; p++) ;  
    Fin Pour  
  
    L <- p - CH ;  
  
    Afficher (" Longueur de la chaine : %d ", L) ;  
  
Fin
```

Exercice 2 :

Ecrire un algo qui lit deux tableaux d'entiers A et B et leurs dimensions N et M au clavier et qui ajoute les éléments de B à la fin de A. Utiliser deux pointeurs PA et PB pour le transfert et afficher le tableau résultant A.

Algorithme de concaténation de tableau :

```
Entier : tabA[100], tabB[50];  
Entier : n, m, i;  
  
Début :  
    afficher(" Entrer une taille pour le tableau A avec n < 50 ");  
    lire(n);  
  
    afficher(" Entrer une taille pour le tableau B avec m < 50 ");  
    lire(m);  
  
    Pour i allant de 1 à n faire :  
        afficher ("élément %d :", i);  
        lire ("%d", tabA+i);  
    Fin Pour  
  
    Pour i allant de 0 à m faire :  
        afficher ("élément %d :", i);  
        lire ("%d", tabB+i);  
    Fin Pour
```

```
Pour i allant de 0 à m faire :  
    afficher("%d", *(tabA+i));  
Fin Pour
```

```
Pour i allant de 0 à m faire :  
    afficher("%d", *(tabB+i));  
Fin Pour
```

```
//Copie de B à la fin de A :  
Pour i allant de 1 à m Faire :  
    *(tabA+n+i) <- *(tabB+i);  
Fin Pour
```

```
n=n+m ;
```

```
Pour i allant de 1 à N Faire;  
    Afficher (" %d ", *(tabA+i));  
Fin Pour
```

```
Fin
```

Exercice 3 :

Ecrire un algo 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. L'algorithme utilisera les pointeurs P1 et P2 pour parcourir le tableau.

Algorithme tassement de tableau :

```
Entier : A[50], L, x, *p1, *p2 ;
```

```
Début :
```

```
    afficher(" Dimension <50 ") ;  
    Lire(N) ;
```

```
    Pour (p1=A ; p1<A+N ; p1++) Faire :  
        afficher("Elément %d", p1-A) ;  
        lire("%d", p1) ;  
    Fin Pour
```

```
    Pour (p1=A ; p1<A+N ; p1++) Faire :  
        afficher("Elément %d", *p1) ;  
    Fin Pour
```

```
    afficher("Entrer X") ;  
    lire(X) ;
```

```
    Pour (p1=p2=A ; p1<A+N ; p1++) Faire :  
        *p2 = *p1 ;  
        Si (*p2 != X) Alors :  
            P2++ ;  
        Fin Si  
    Fin Pour
```

N = p2-A

Pour (p1=A ; p1<A+N ; p1++) Faire :
 afficher("%d", *p1) ;
Fin Pour

Fin

Exercice 4 :

Ecrire un algorithme qui range les éléments d'un tableau A du type int dans l'ordre inverse. L'algorithme utilisera des pointeurs P1 et P2 et une variable numérique AIDE pour la permutation des éléments.

Algorithme inversion:

Variables :

Définir : MAX = 20

Entier : tab[MAX], size, tmp, i, *p1, *p2 ;

Debut :

Tant que (size < 0 **OU** size > MAX) Faire:
 Afficher("Entrez une taille <%d:", MAX);
 Lire(size);
Fin tant que

Pour i allant de 0 à size Faire:
 Afficher("tab[%d]: ", i)
 Lire(tab+i) ;
Fin pour

Afficher("Tableau :")
Pour i allant de 0 à size Faire:
 Afficher("[%d] ", *(tab+i));
Fin pour

Pour j allant de n à 0 Faire:
 P1 <= tab+i
 P2 <= (tab+size-i-1);
 tmp <= *p1
 *p1 <= *p2
 * p2 <= tmp
Fin pour

Afficher("Tableau inversé :")
Pour i allant de 0 à size Faire:
 Afficher("[%d] ", *(tab+i));
Fin pour

Fin