

Exercice 1 : (4.5 pts)

Écrivez une procédure qui prend en entrée deux tableaux t1 et t2, ainsi que leurs tailles respectives n1 et n2. Cette procédure doit renvoyer un nouveau tableau t et sa taille n. Ce tableau doit contenir les éléments communs aux deux tableaux, c'est-à-dire l'intersection des deux tableaux.

اكتب إجراء يأخذ جدولين t1 و t2 كمدخلات، بالإضافة إلى أحجامهما n1 و n2. يقوم هذا الإجراء بإرجاع جدول جديد t بحجم n، يحتوي على العناصر المشتركة في كليهما، أي الجزء المشترك بينهما.

Exemple : t1={1,4,3,12,5} t2={12,8,22,4,9} → t={4,12}

<code>void * interTab (int *t1, int n1, int *t2, int n2, int *t, int *n,){</code>
<code>int i, j;</code>
<code>t= (int *)malloc(n1 * sizeof(int));</code>
<code>*n=0 ;</code>
<code>for (i = 0; i < n1; i++)</code>
<code>for (j = 0; j < n2; j++)</code>
<code>if (t1[i] == t2[j]) {</code>
<code> t[*n] = t1[i];</code>
<code> (*n)++;</code>
<code>}</code>
<code>}</code>
<code>t = (int *)realloc(t, (*n) * sizeof(int));</code>
<code>}</code>

Exercice 2 : (3.5 pts)

Écrire une fonction récursive permettant de savoir si une chaîne de caractère est palindrome ou non.

Un **palindrome** est une séquence de caractères qui reste identique lorsqu'elle est lue de gauche à droite ou de droite à gauche

اكتب دالة تراجعية بلغة C تسمح بمعرفة ما إذا كانت سلسلة الأحرف متناظرة أم لا.

Exemple : radar, eve, تحت

<code>int isPalindrome(char* s, int start, int end) {</code>
<code> if (start>= end) return 1 ;</code>
<code> if (s[start]!= s[end]) return 0 ;</code>
<code> return isPalindrome(s, start+1, end-1) ;</code>
<code>}</code>

Exercice 3 : (4.75 pts)

Écrire une fonction qui prend en entrée deux listes d'entiers et renvoie une nouvelle liste contenant uniquement les éléments qui existent dans la première liste mais pas dans la deuxième liste.

اكتب دالة تأخذ كمدخلات قائمتين من الأعداد الصحيحة وتعيد قائمة جديدة تحتوي فقط على العناصر الموجودة في القائمة الأولى ولكن ليس في القائمة الثانية.

Exemple : entrées L1->8->3->5->2->4->fin L2->2->7->3->fin résultante : 8->5->4->fin

List diffList (List L1, List L2){
List t, L=NULL;
while(L1){
t= L2;
exist=0;
while(t && !exist){
if(t->data== L1->data)
exist=1 ;
t= t->next;
}
if(!exist)
append_end(L, L1->data)
L1= L1->next
}
return L ;
}

Exercice 4 : (7.25 pts)

Écrire une procédure qui prend une liste chaînée et qui échange chaque élément en position impaire (pas seulement les valeurs) avec le nœud qui le suit directement.

اكتب إجراء يأخذ قائمة مرتبطة ويقوم بتبادل كل عنصر في موضع فردي (ليس القيم فقط) مع العقدة التي تليه مباشرة.

Exemple : la liste entrée 5->9->2->7->8->fin la liste résultante : 9->5->7->2->8->fin

void OddSwap (List &L){
Node *p,*t ;
if(L ==NULL L->next ==NULL) return;
p=L;
L=L->next;
p->next=L->next;
L->next=p;
t= p->next ;
while(t !=NULL && t->next !=NULL) {
p->next =t->next ;
p =p->next ;
t->next =p->next ;
p->next =t ;
p =t ;
t =t->next ;
}
}