



Niveau : 1<sup>o</sup> année informatique

Matière : ASD2

Nom اللقب : -----

## Corrige type 02

Année universitaire : 2022/2023

Durée : 1h :30m

Groupe الفوج : -----

Prénom الاسم : -----

### Exercice 1 : (5 pts)

Écrire une fonction prenant deux arguments, un entier x et un entier k, et renvoyant un tableau contenant les k derniers diviseurs de x.

اكتب دالة تأخذ وسيطين ، عددًا صحيحًا x وعددًا صحيحًا k ، وتعيد جدولًا يحتوي على آخر k قواسم لـ x.

Exemple : x=24 k=4 t={24,12,8,6}

<b>int* kDiv (int x, int k){</b>	
<b>int *t, i, d ;</b>	0.75
<b>t =(int*)malloc(k*sizeof int) ;</b>	1
<b>for(i=0,d=x;d&gt;0&amp;&amp;i&lt;k;d--)</b>	1.25
<b>if(x%d==0){</b>	0.5
<b>t[i]=d ;</b>	0.5
<b>i++;</b>	0.5
<b>}</b>	
<b>return t ;</b>	0.5
<b>}</b>	

### Exercice 2 : (5 pts)

Écrire une fonction prenant en entrée deux listes chaînées (sans doublons) et renvoyant une troisième liste chaînée qui contient les éléments communs aux deux listes (c'est-à-dire l'intersection des deux listes).

اكتب الدالة التي تأخذ قائمتين مرتبطتين (بدون تكرارات) وترجع قائمة ثالثة تحتوي على العناصر المشتركة في كلتا القائمتين (أي تقاطع القائمتين).

Exemple : les listes entrées a :1->4->3->12->5->fin b :12->8->22->4->9 la liste résultante : 12->4->fin

<b>List intersectList (List a, List b) {</b>	
<b>List L=NULL,t;</b>	0.5
<b>while(a) {</b>	0.5
<b>t=b;</b>	0.5
<b>while(t)</b>	0.5
<b>if(a-&gt;data==t-&gt;data) {</b>	0.5
<b>append_end(L,a-&gt;data);</b>	0.5
<b>t=NULL;</b>	0.5
<b>} else</b>	
<b>t=t-&gt;next;</b>	0.5
<b>a=a-&gt;next;</b>	0.5
<b>}</b>	
<b>return L;</b>	0.5
<b>}</b>	

### Exercice 3 : (5 pts)

Écrire une procédure qui prend une liste chaînée et qui échange la valeur de chaque élément en position impaire avec la valeur du nœud qui le suit directement.

اكتب إجراء يأخذ قائمة مرتبطة ويقوم بتبادل قيمة كل عنصر في موضع فردي مع قيمة العنصر الذي يليه مباشرةً.

Exemple : la liste entrée

5->9->2->7->8->fin

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

<b>void OddSwap (List L){</b>	
int t ;	0.5
if(L ==NULL    L->next ==NULL) return;	1
while(L !=NULL && L->next !=NULL) {	1
t= L->data;	0.5
L->data = L->next ->data;	0.5
L->next ->data =t ;	0.5
L=L->next ->next ;	1
}	
}	

### Exercice 4 : (5 pts)

Écrire une fonction récursive permettant de savoir si un nombre x est premier ou non.

En utilisant cette fonction écrire une fonction récursive pour afficher tous les nombres premiers inférieur a « n ».

اكتب دالة تراجعية لمعرفة ما إذا كان العدد x عددًا أوليًا أم لا.

باستخدام هذه الدالة ، اكتب دالة تراجعية لعرض جميع الأعداد الأولية الأقل من "n".

<b>int isPrime(int x, int d) {</b>	
if (x==d ) return 1 ;	1
if (x%d== 0) return 0 ;	1
return isPrime(x, d+1) ;	1
}	
<b>void afficher (int n){</b>	
if (n>1) {	0.5
if (isPrime(n, 2))	0.5
printf("%d\n",n) ;	0.5
afficher (n-1);	0.5
}	
}	