

QROC**Module A31TTN**

Structures de données

Aucun document n'est autorisé, les réponses sont à donner sur le présent document.

Exercice 1 – Récursivité (4 points)

- 1) Ecrire deux fonctions, dont une récursive, qui calculent le factoriel d'un nombre n ($n!$).

<i>/*Version itérative*/</i>	<i>/*Version récursive*/</i>
------------------------------	------------------------------

- 2) Evaluer la complexité de chacune des deux fonctions, expliquer.

--

Exercice 2 – Listes chaînées (8 points)

Un étudiant souhaite mettre en place un programme en C qui a pour objectif de concaténer deux listes chaînées. Par exemple, à partir des listes $T1=\{1,4\}$ et $T2=\{2,3\}$ retourner puis afficher la liste $\{1,4,2,3\}$, le résultat de la concaténation des listes $T1$ et $T2$. Il a écrit une partie du code donnée ci-après. Il demande votre aide pour compléter son code.

```
#include<stdio.h>
#include<conio.h>
#include<stdlib.h>

typedef struct element
{
    int val;
    struct element *suivant;
}maillon;
typedef maillon* liste;
```


```

/* fonction X */
liste X(liste l,int valeur) {
    liste nv=(liste)malloc(sizeof(maillon));
    nv->val=valeur;
    if(l==NULL) {
        l=nv;
        nv->suivant=NULL;
    }
    else {
        nv->suivant=l;
        l=nv;
    }
    return l;
}
/* fonction concat */
liste concat(liste T1,liste T2)
{
    ....
    /* Partie du code à compléter */
    ...
}
/* fonction Y */
void Y(liste l)
{
    liste tmp=l;
    while(tmp!=NULL)
    {
        printf("%d->",tmp->val);
        tmp=tmp->suivant;
    }
}
/* Programme principal */
int main()
{
    liste T1=NULL,T2=NULL;
    T1=X(T1,4);
    T1=X(T1,1);
    T2=X(T2,3);
    T2=X(T2,2);
    Y(T1);
    T1=concat(T1,T2);
    Y(T1);
    return 0;
}

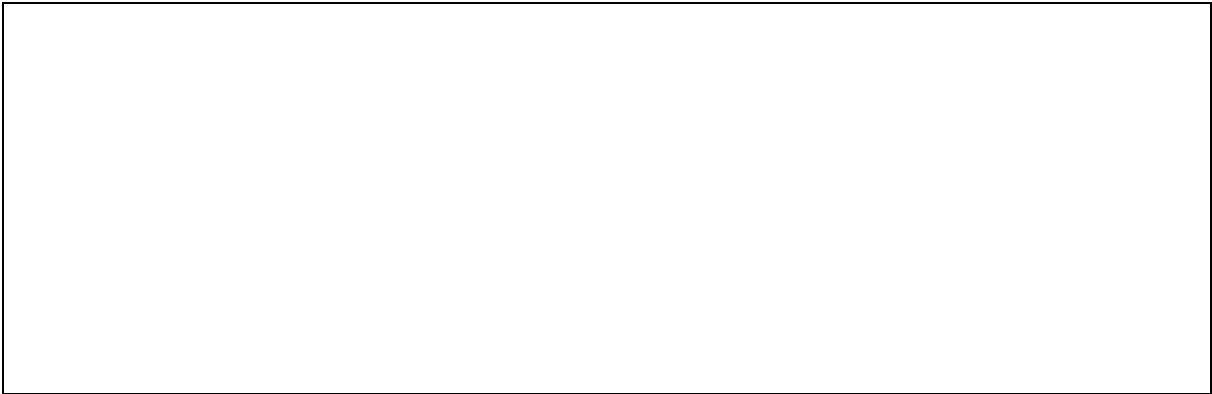
```

- 1) Expliquez ce que font les fonctions X et Y données dans le code ci-dessus. Donnez des noms significatifs à ces fonctions.

- 2) Compléter la partie manquante du code (*pas besoin de réécrire la totalité du code*).
N'oubliez pas de traiter les cas où une des listes est vide.



- 3) Ecrire une fonction qui renvoie le nombre d'éléments d'une liste chaînée.



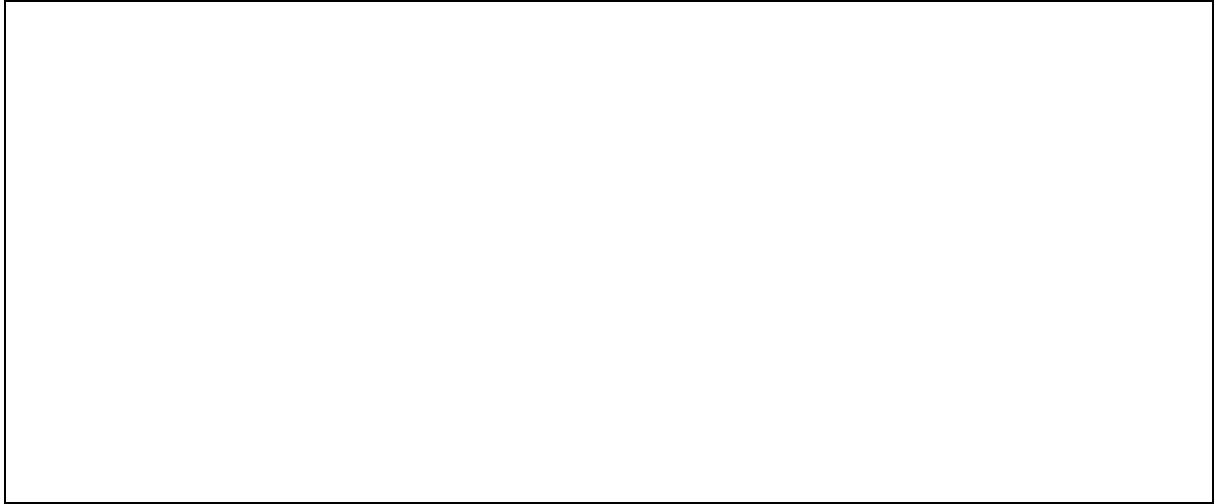
- 4) Ecrire une fonction qui vérifie si une liste chaînée est triée.



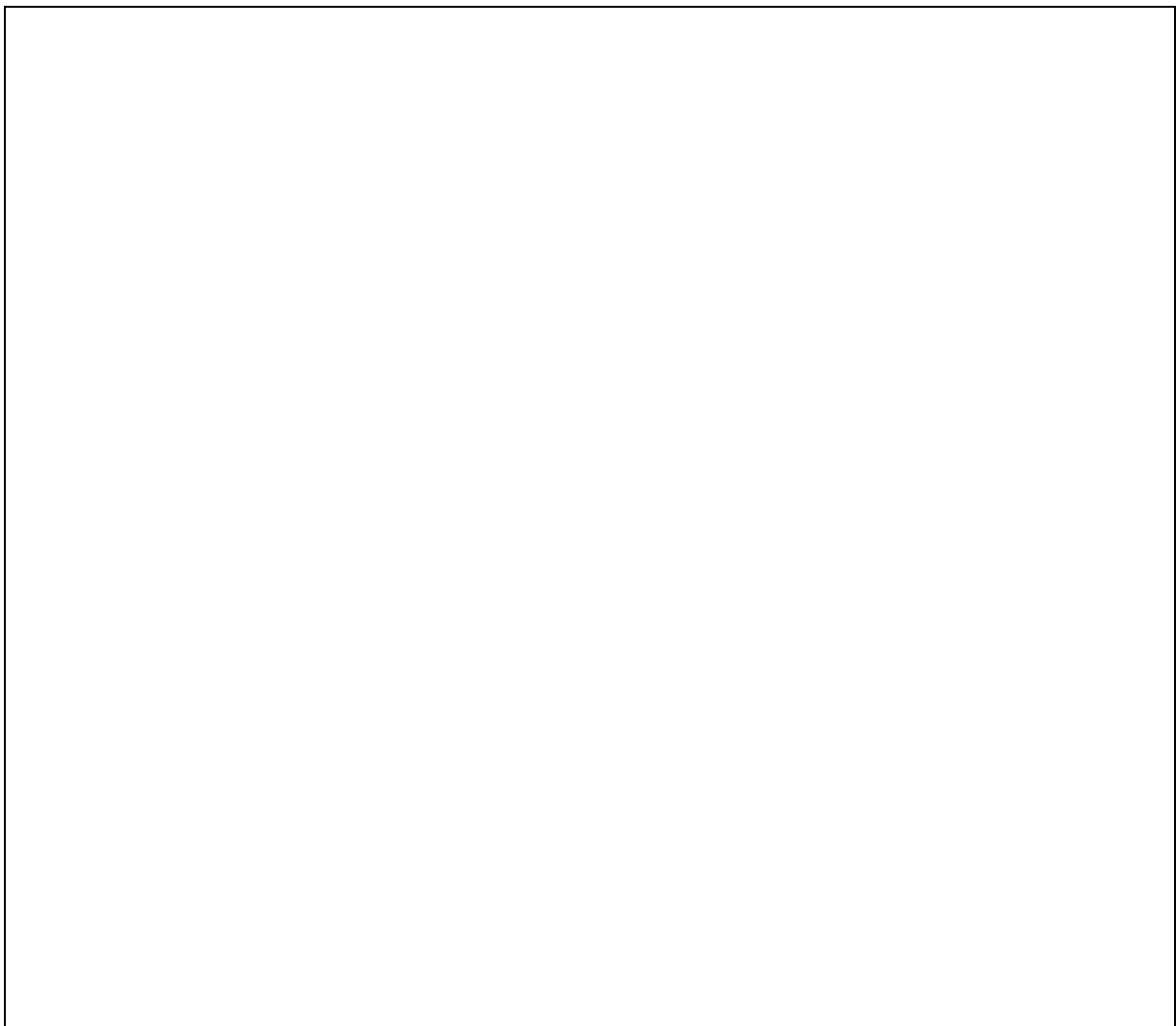
Exercice 3 – Les arbres binaires (8 points)

On considère la suite de nombres suivante : 6, 10, 11, 8, 9, 20, 30, 13, 15, 14.

- 1) Partant d'un arbre binaire de recherche vide, on ajoute successivement les valeurs de la suite ci-dessus. Donnez en justifiant votre raisonnement l'arbre résultat.



- 2) Vérifier si l'arbre obtenu est équilibré, s'il n'est pas équilibré, équilibrez-le. Expliquez les opérations de rotation effectuées. (*On recommencera l'insertion dans un arbre vide*).



3) Proposez une structure de donnée qui permet une représentation chaînée de cet arbre.

4) Sur l'arbre obtenu, calculer manuellement sa hauteur. Proposez une fonction qui retourne la hauteur d'un arbre.

5) Proposez une fonction qui permet de tester si un arbre binaire donné en paramètre est un arbre binaire de recherche.

Bonne chance.